

РОЧЕВ К. В.
АНАЛИЗ БЫСТРОДЕЙСТВИЯ ТИПОВЫХ ОПЕРАЦИЙ ЯЗЫКА C#
НА ПЛАТФОРМАХ DOT.NET И MONO
УДК 004:4'2, ВАК 05.13.18, ГРНТИ 50.41.01

Анализ быстродействия типовых
операций языка C#
на платформах DOT.NET и Mono

Standard operations performance
analysis of the C# language
on platforms DOT.NET and Mono

К. В. Рочев

К. В. Рочев

Ухтинский государственный
технический университет, г. Ухта

Ukhta State Technical University,
Ukhta

Статья посвящена изучению быстродействия часто используемых функций стандартных классов языка C# в разных окружениях, таких как WPF, Windows forms, Unity и ASP.NET. Реализован асинхронный механизм инструментальной оценки быстродействия участков кода. Рассмотрены несколько версий фреймворка, включая Mono, Core и традиционный .NET Framework, чтобы выявить разницу в скорости выполнения тех или иных функциональных возможностей платформ.

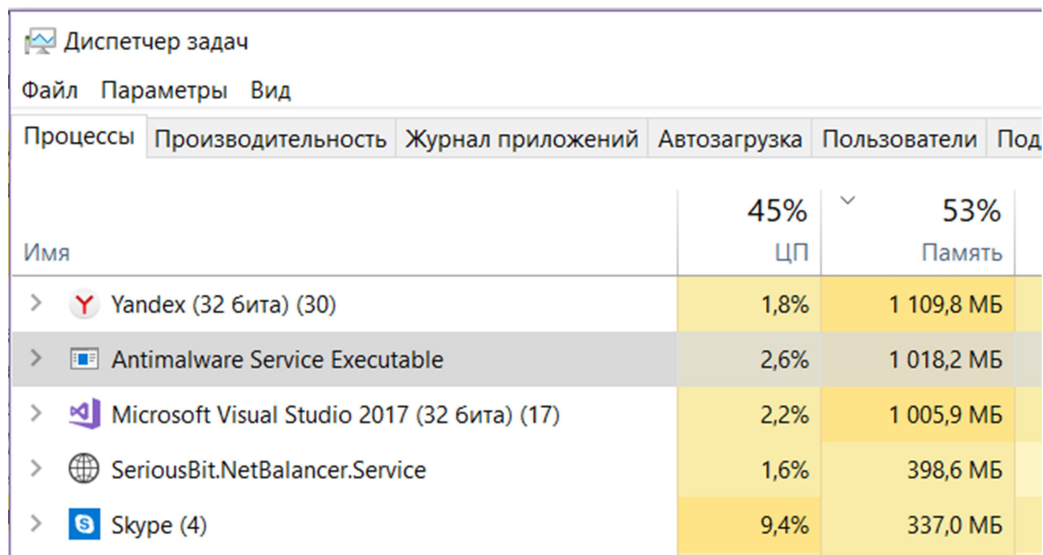
The article is devoted to the study of C# language frequently used functions performance in different environments, such as WPF, Windows forms, Unity and ASP.NET. Implemented an asynchronous mechanism tool of evaluating the performance of code regions. Several versions of the framework, including Mono, Core and the traditional .NET Framework, are reviewed to determine the difference in the speed of execution of certain platform functionalities.

Ключевые слова: быстродействие, оптимизация кода, производительность, C#, WPF, Windows Forms, DOT.NET Framework, Unity, Core

Keywords: performance, code optimization, C#, WPF, Windows forms, DOT.NET Framework, Unity, DOT.NET core

Введение

Оптимизации быстродействия программного обеспечения многими разработчиками в настоящее время уделяется недостаточно внимания. Это хорошо заметно по медленной работе некоторых современных приложений и их большим размерам. В качестве примера можно привести несколько самых распространенных по назначению приложений, реализующих такие функции как набор текстов, просмотр страниц в сети Интернет, голосовая связь (рис. 1), с которым справлялись и машины конца прошлого века, обладавшие в десятки и сотни раз меньшими мощностями, чем нынешние [1].



Диспетчер задач

Файл Параметры Вид

Процессы Производительность Журнал приложений Автозагрузка Пользователи Под

Имя	45% ЦП	53% Память
> Yandex (32 бита) (30)	1,8%	1 109,8 МБ
> Antimalware Service Executable	2,6%	1 018,2 МБ
> Microsoft Visual Studio 2017 (32 бита) (17)	2,2%	1 005,9 МБ
> SeriousBit.NetBalancer.Service	1,6%	398,6 МБ
> Skype (4)	9,4%	337,0 МБ

Рисунок 1. Пример объёмов памяти, занимаемых современным программным обеспечением

Проблема кроется, зачастую, в использовании излишне тяжелых библиотек и компонентов, которые тяжелы из-за того, что сами используют другие библиотеки и это дерево уходит своими корнями глубоко в историю. Не меньшее влияние оказывает и неправильное применение структур данных, не учитывающие сложности алгоритмов при обработке множеств элементов. Элементарные операции и типовые функции из стандартных библиотек, на первый взгляд, влияют на быстродействие в меньшей степени, однако они, ввиду частого использования тоже, порой, вносят своё воздействие в изменение быстродействия. Особенно это становится заметно при обработке больших объемов данных.

Данная статья посвящена изучению быстродействия элементарных операций и часто используемых функций языка C# в разных окружениях, таких как WPF, Windows forms, Unity. Мы рассмотрим несколько версий фреймворка и видов проекта, чтобы увидеть, есть ли разница в скорости выполнения того или иного функционала.

Для изучения напишем небольшой тестовый класс, разместим его в переносимой библиотеке и будем ее подключать в разные среды выполнения. Компактная переносимая библиотека классов, подключаемая к разным средам выполнения с исходным кодом размещена в открытом доступе на сервисе Bitbucket [2].

Методология и реализация тестового окружения

Основной функционал класса, реализованного для изменения быстродействия, следующий:

1) поток, постоянно выполняющий изучаемую функцию из её делегата *private Action TestAction*, что позволяет частично обойти оптимизацию повторяющихся операций механизмами .Net Framework-a;

2) функция замера, принимающая делегат – замеряет количество выполнений этого делегата потоком в течение 1 миллисекунды;

3) функция подсчета – накапливает результаты одинаковых замеров в словаре для последующего устранения пиковых результатов, усреднения, вычисления медианного значения;

4) механизм минимизации вызова сборщика мусора и замера частоты его во время тестов.

Класс тестировщика производительности представлен ниже:

```
public class TimeTestAsync
{
    public int TimeMilliseconds = 1;

    private Action TestAction = () => { };
    private Thread Thread;
    public int Count = 0;

    public readonly Dictionary<string, List<TimeResult>> Results =
        new Dictionary<string, List<TimeResult>>();

    private void Init()
    {
        if (Thread != null) return;
        Thread = new Thread(TestFunk) { IsBackground = true };
        Thread.Start();
    }

    ~TimeTestAsync() => Stop();

    public void Zamer(string info, Action action)
    {
        Init();
        if (!Results.ContainsKey(info))
            Results.Add(info, new List<TimeResult>());
        var z = Zamer(action);
        Results[info].Add(z);
    }

    private DateTime _start;

    private TimeResult Zamer(Action action)
    {
        var gc = GC.CollectionCount(0);
        TestAction = action;
        Count = 0;
        _start = DateTime.UtcNow;
        Thread.Sleep(TimeMilliseconds);
        var end = DateTime.UtcNow.Subtract(_start);
        return new TimeResult()
        {
            Count = Count,
            Time = end,
            GC = GC.CollectionCount(0) - gc
        };
    }

    private void TestFunk()
    {
        while (true)
        {
            TestAction.Invoke();
            ++Count;
        }
    }
}
```

```

    }
}

public void Stop()
{
    Thread?.Abort();
    Thread = null;
}
}

```

Результат замера выглядит следующим образом:

```

public class TimeResult
{
    public int Count;
    public TimeSpan Time;
    public int GC;

    public double Nanoseconds()
    {
        return Time.TotalMilliseconds / Count * 1000000;
    }

    public override string ToString()
    {
        return "\t" + Nanoseconds() + "\t" + Count;
    }
}

```

По времени замера и количеству операций он определяет время выполнения тестируемой функции в наносекундах.

И, собственно, применение этого класса замеров возможно, например, в таком виде:

```

public class Tests
{
    readonly TimeTestAsync Tester = new TimeTestAsync();
    private const string Br = "\t";
    private float ClassProperty { get; set; }
    static float StaticProperty { get; set; }
    static float StaticField = 0;
    float ClassField = 0;
    float ClassField2 = 0;
    string ClassStr = "";
    bool ClassBool = false;
    private const int Min = 50, Max = 100;

    ...

    public string Test()
    {
        var localRandom = new Random();

        for (int i = 0; i <= 10; i++)
        {
            ClassStr = "";
            ClassStringBuilder = new StringBuilder();
            ClassField = localRandom.Next(Min, Max);

            ...

            GC.Collect();
            Tester.Zamer("() => { ClassField++; }", () => { ClassField++; });
            Tester.Zamer("() => ClassStr = \"S1\"", () => ClassStr = "S1");
            Tester.Zamer("() => ClassStr = \"S1\" + ++ClassField",
                () => ClassStr = "S1" + ++ClassField);
            GC.Collect();

```

```

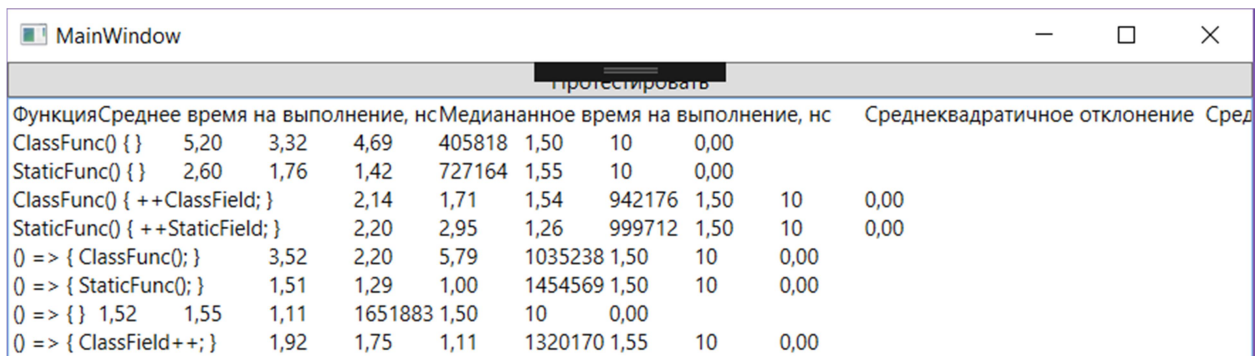
...
        if (i == 0) Tester.Results.Clear();
        StaticField += localField;
    }

    Tester.Stop();
    var tempCount = ClassField + StaticField;

    string s = $"Функция{Br}" +
        $"Среднее время на выполнение, нс{Br}" +
        $"Медиананное время на выполнение, нс{Br}" +
        $"Среднеквадратичное отклонение{Br}" +
        $"Среднее к-во запусков за тест, раз{Br}" +
        $"Среднее Время теста, мс{Br}" +
        $"К-во тестов, раз{Br}" +
        $"Среднее к-во вызовов сборки мусора на тест, раз\n";
    foreach (var r in Tester.Results)
    {
        var withResults = r.Value.Where(x => x.Count > 0).ToArray();
        var nano = withResults.Select(x => x.Nanoseconds()).ToList();
        s += $"{r.Key}{Br}" +
            $"{nano.Average(x => x):F2}{Br}" +
            $"{nano.OrderBy(x => x).ToArray()[nano.Count / 2]:F2}{Br}" +
            $"{StandardDeviation(nano):F2}{Br}" +
            $"{withResults.Average(x => (double)x.Count):F0}{Br}" +
            $"{withResults.Average(x => x.Time.TotalMilliseconds):F2}{Br}" +
            $"{withResults.Length}{Br}" +
            $"{withResults.Average(x => (double)x.GC):F2}\n";
    }
    s += "\n\n" + tempCount; // + "\n\n" + Tester.GcCount
    return s;
}

```

Простейший интерфейс позволяет скопировать результаты в Excel и там их обработать:



Функция	Среднее время на выполнение, нс	Медиананное время на выполнение, нс	Среднеквадратичное отклонение	Сред
ClassFunc() {}	5,20	3,32	4,69	405818
StaticFunc() {}	2,60	1,76	1,42	727164
ClassFunc() { ++ClassField; }		2,14	1,71	1,54
StaticFunc() { ++StaticField; }		2,20	2,95	1,26
() => { ClassFunc(); }	3,52	2,20	5,79	1035238
() => { StaticFunc(); }	1,51	1,29	1,00	1454569
() => { }	1,52	1,55	1,11	1651883
() => { ClassField++; }	1,92	1,75	1,11	1320170

Рисунок 2. Результат измерения в простом окне WPF для последующего копирования в Excel

Относительные результаты измерения быстродействия по рассмотренным группам операций оценивались относительно WPF по следующей формуле:

$$X = \frac{\sum S_t \cdot 2}{\sum S_t + S_{WPF}} \quad (1)$$

где X – относительное быстродействие; S_t – Результат на рассматриваемой платформе; S_{WPF} – Результат на WPF.

Результаты измерений

Измерение проводилось на ноутбуке ASUS X556UQ: i7-7500U, 2.7 GHz, 20Г ОЗУ, Windows 10 x64.

Для оценки быстродействия реализованного тестового окружения были выбраны такие операции, как обращение к функциям, полям и свойствам класса (табл. 1 и 2). Как можно заметить, быстродействие тестовой инфраструктуры примерно сопоставимо с обычным обращением к функции или переменной. Кроме того, несмотря на предпринятые меры по усложнению оптимизации вычислений, в простых операциях замечен разгон при выполнении нескольких однотипных замеров подряд. В дальнейшем было принято решение рассматривать и сопоставлять результаты по медианному времени, поскольку того, что оно более стабильно, чем среднее, т. к. не подвержено влиянию исключительных случаев. Хотя, как ни странно, из-за особенностей разброса результатов (и сдвига в большую сторону при четном количестве тестов) в некоторых замерах медианный результат получен больше среднего.

Таблица 1. Результаты тестовых замеров на примере проекта WPF в Release-режиме

Функция	Среднее время на выполнение, нс	Медианное время на выполнение, нс	Среднеквадратичное отклонение	Среднее к-во запусков за тест, раз	Среднее Время теста, мс	К-во тестов, раз	Среднее к-во сборки мусора на тест, раз
ClassFunc() {}	2,02	1,97	0,23	736911	1,48	10	0
StaticFunc() {}	1,22	1,12	0,48	1268418	1,45	10	0
ClassFunc() { ++ClassField; }	1,74	2,57	1	1241860	1,57	10	0
StaticFunc() { ++StaticField; }	1,34	1,15	0,9	1617269	1,5	10	0
() => { ClassFunc(); }	0,87	0,76	0,5	2067692	1,41	10	0
() => { StaticFunc(); }	0,73	0,55	0,52	2737093	1,55	10	0
() => { }	0,61	0,54	0,43	3381908	1,6	10	0
() => { ClassField++; }	0,46	0,42	0,24	3979307	1,55	10	0
() => { ++ClassField; }	1,02	0,33	1,1	3631160	1,49	10	0
() => { ++localField; }	1,7	2,58	1,18	2409092	1,47	10	0
() => { ++StaticField; }	1,58	1,46	0,99	2119760	1,5	10	0
() => { ++ClassProperty; }	1,45	1,3	0,89	1863567	1,55	10	0
() => { ++StaticProperty; }	1,98	1,02	1,92	1944740	1,46	10	0

Далее приведем результаты замеров в разных режимах выполнения (табл. 2), поскольку на второй фазе компиляции при создании релиз-приложения используются дополнительные механизмы оптимизации [3].

Таблица 2. Замеры в WPF в разных режимах выполнения

Функция	WPF DEBUG с отладчиком, нс	WPF DEBUG без отладчика, нс	WPF RELEASE с отладчиком, нс	WPF RELEASE без отлад- чика, нс
ClassFunc() {}	4,83	3,68	2,39	2,10
StaticFunc() {}	4,65	3,10	2,90	1,72
() => {}	4,08	3,15	3,07	1,64
() => { ClassFunc() {} }	7,66	4,20	2,17	1,55
() => { StaticFunc() {} }	7,22	4,82	2,29	1,17
() => { ++LocalInt; }	16,20	4,28	2,12	1,51
() => { ++ClassIntField; }	5,95	3,77	2,33	1,55
() => { ++StaticIntField; }	4,52	3,09	2,16	1,36
() => { ++ClassIntProperty; }	20,18	7,80	1,98	1,61
() => { ++StaticIntProperty; }	11,46	7,47	1,80	1,49

Как можно заметить, элементарные операции довольно существенно оптимизируются при переводе проекта в релиз. Устраняются лишние сложности вызова свойств, и они начинают работать также быстро, как обычные поля, удаляются вызовы пустых функций. Расхождения в замерах быстродействия становятся обусловлены в большей степени случайными флуктуациями. При этом оптимизатор настолько хорош, что можно заметить постепенное ускорение быстродействия для похожих операций, несмотря на их вызов через делегаты.

Далее для сопоставления фреймворков будем рассматривать наиболее актуальный режим запуска – Release.

На рисунке 3 приведены результаты замеров выполнения указанных функций в релиз-проектах на WPF, Windows Forms и Unity. Как можно заметить, WPF и Windows Forms показывают примерно одинаковые результаты (в среднем по рассмотренным операциям формы медленнее на 6 %) ввиду того, что обе платформы реализованы на классическом .Net (при этом, в дебаг-режиме разница между ними более существенная). В то же время, на Unity некоторые операции производятся с существенной разницей в скорости ввиду того, что основаны на MONO-Framwork-e (Unity, в среднем, медленнее на 220 %). В .Net Core реализации заметна не менее ощутимая разница в быстродействии, как в большую, так и в меньшую сторону по разным операциям (в среднем по рассмотренным функциям на 77 % медленнее). Однако выборка функций не является достаточно презентабельной и не даёт полномочий судить о производительности того или иного фреймворка в целом.

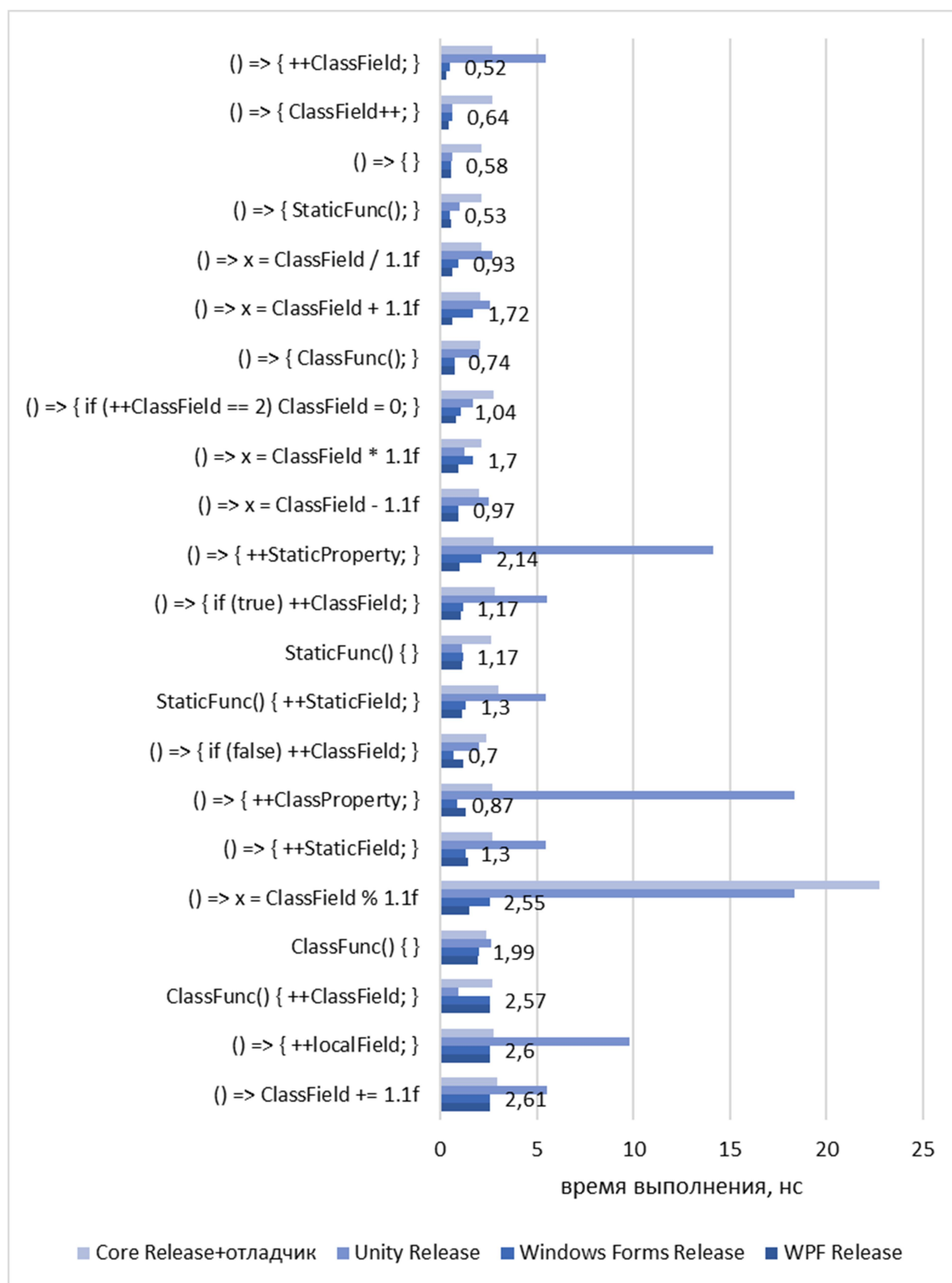


Рисунок 3. Быстродействие обращения к методам, полям и свойствам в различных окружениях

Здесь заметны некоторые сложности с обращением к свойствам в Unity, а также заметно более медленный расчет остатка от деления в Unity и .Net Core – примерно в 10 раз медленнее, чем в обычном .Net, при том, что остальные операции выполняются примерно с той же скоростью. В .Net Core тестовое

окружение (вызов делегата) выполнялось медленнее, учитывая это можно отметить, что остальные обращения и вычисления в нём производились с той же скоростью, что и в обычном .Net.

Быстродействие математических операций

Рассмотрим несколько наиболее популярных математических операций стандартного .Net класса Math (табл. 3 и рис. 4).

Таблица 3. Замеры быстродействия математических операций в различных окружениях

Функция	WPF	Windows Forms	Unity	Core	Unity/WPF	Core/WPF
() => Math.Atan(++ClassField) }	54,8	54,87	27,02	2,63	66 %	9 %
() => Math.Atan2(++ClassField, ClassField) }	53,79	53,94	35,93	30,31	80 %	72 %
() => Math.Pow(++ClassField, 2)	45,03	45,14	52,42	2,78	108 %	12 %
() => Math.Pow(++ClassField, 1.4)	44,96	45,85	52,43	2,63	108 %	11 %
() => Math.Round(65.5633, 2)	19,79	21,62	30,31	13,99	121 %	83 %
() => localRandom.Next(10, 10000)	19,44	20,03	15,64	14,04	89 %	84 %
() => ClassField = localRandom.Next(10)	17,03	16,37	16,07	15,82	97 %	96 %
() => Math.Round(0d, 2)	16,01	16,22	10,36	10,88	79 %	81 %
() => Math.Round(0m, 2)	15,39	15,39	17,1	15,74	105 %	101 %
() => localRandom.Next(10)	15,01	15,36	15,12	13,92	100 %	96 %
() => localRandom.NextDouble()	12,65	13,68	14,57	12,14	107 %	98 %
() => Math.Ceiling(++ClassField) }	7,23	7,28	10,3	2,83	118 %	56 %
() => Math.Floor(++ClassField) }	6,69	6,8	10,04	2,72	120 %	58 %
() => Math.Sin(++ClassField) }	3,42	3,06	56,25	2,78	189 %	90 %
() => Math.Sqrt(++ClassField) }	2,6	2,6	9,84	2,75	158 %	103 %
() => Math.Abs(++ClassField) }	2,59	2,59	17,66	2,94	174 %	106 %
() => Math.Log(++ClassField) }	2,21	2,21	14,89	8,94	174 %	160 %
() => Math.Cos(++ClassField) }	1,51	1,54	46,39	2,63	194 %	127 %
() => Math.Tan(++ClassField) }	1,49	2,17	27,49	2,65	190 %	128 %

Как можно заметить, различные математические операции выполняются в разных средах разработки с существенными отличиями в быстродействии. Например, Sin, Cos, корень, логарифм и получение модуля намного медленнее работают в Unity, чем в других версиях фреймворка, да и в среднем математика в Unity работает несколько медленнее. Хотя, например, арктангенс вычисляется быстрее.

Интересен тот факт, что возведение в степень с помощью функции Pow работает с одной скоростью для целых и дробных степеней и на пару порядков медленнее умножения (за исключением .Net Core, где оно приближается по скорости к простым арифметическим операциям).

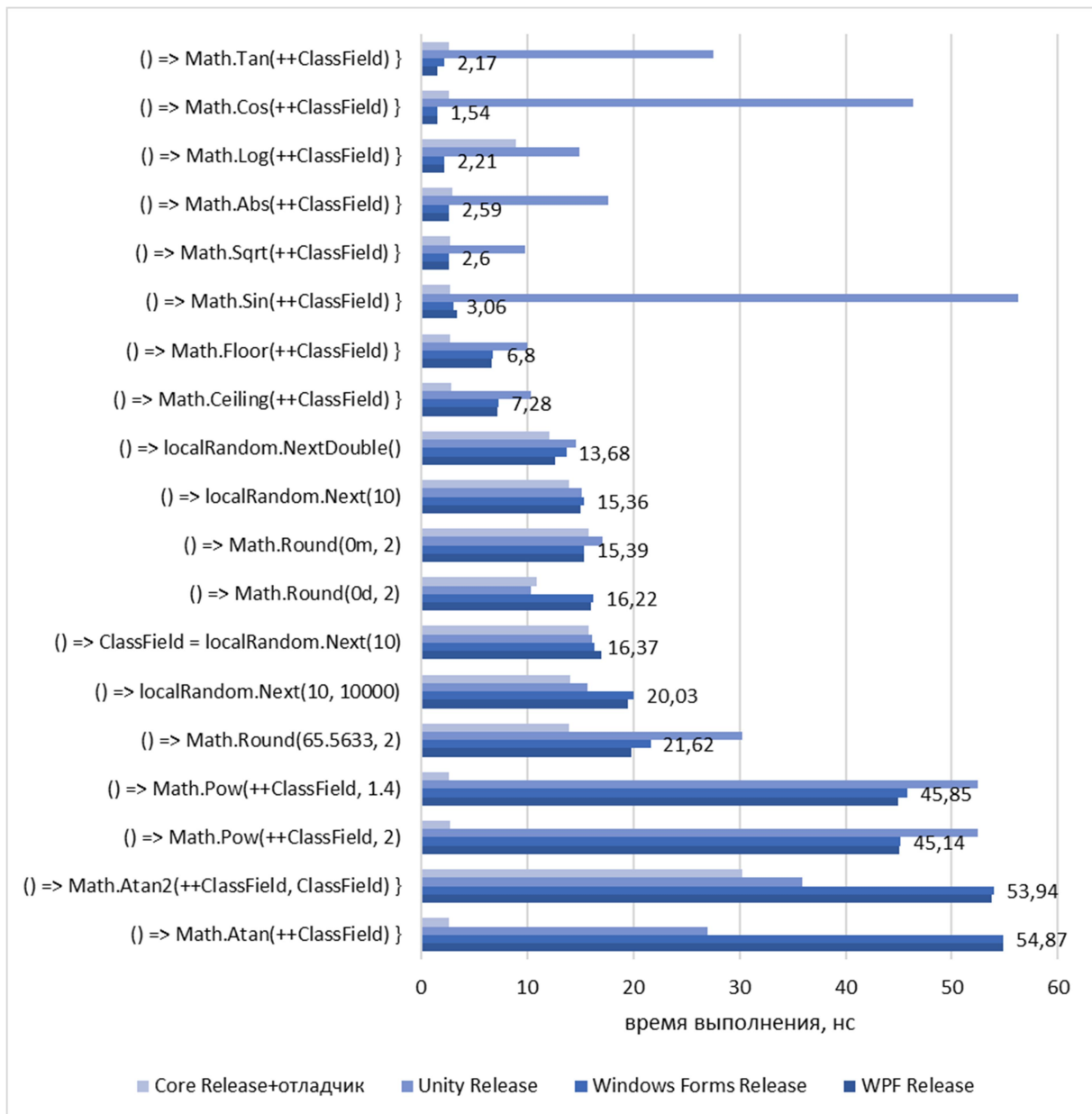


Рисунок 4. Замеры быстродействия математических операций в различных окружениях

Быстродействие функций работы с коллекциями

Приведем результаты исследования работы с коллекциями кратко на примере массивов из 1 и 1000 элементов и таких часто используемых функций, как Contains – поиск элемента, FirstOrDefault – аналогичная функция в LINQ-расширениях, Count – подсчёт с помощью LINQ. ExistElement – случай для элемента, который присутствует в массиве, а NotExistElement – соответственно для несуществующего элемента.

Таблица 4. Замеры быстродействия работы с массивами в различных окружениях

Функция	WPF	Windows Forms	Unity	Core	Unity/WPF	Core/WPF
Массив из 1 элемента						
Contains(ExistElement)	42	41	58	31	116 %	85 %
Contains(NotExistElement)	42	43	58	32	116 %	86 %
FirstOrDefault(x => x == ExistElement)	29	29	72	32	143 %	106 %
FirstOrDefault(x => x == NotExistElement)	34	35	74	35	137 %	101 %
FirstOrDefault()	36	23	17	42	63 %	108 %
Count(x => x == ExistElement)	32	33	74	26	140 %	90 %
Count(x => x == NotExistElement)	36	34	77	29	136 %	88 %
Массив из 1000 элементов						
Contains(ExistElement)	1063	1120	15036	840	187 %	88 %
Contains(NotExistElement)	1835	1826	47786	1257	193 %	81 %
FirstOrDefault(x => x == ExistElement)	4685	5470	4664	6318	100 %	115 %
FirstOrDefault(x => x == NotExistElement)	8299	8643	16152	8592	132 %	102 %
FirstOrDefault()	37	38	17	42	64 %	106 %
Count(x => x == ExistElement)	9217	9229	16667	9398	129 %	101 %
Count(x => x == NotExistElement)	9281	8351	16215	8786	127 %	97 %

Как можно заметить, различия в скорости выполнения присутствуют и увеличиваются с увеличением объёма массива. Практически во всех рассмотренных случаях Mono реализация Unity несколько уступает по скорости.

Что интересно, проверка наличия элемента с помощью встроенной функции Contains проходит быстрее, чем с помощью LINQ расширения примерно в 3–5 раз в классическом и Core фреймворках и в 3 раза медленнее Юнити на 1000 элементах. А на 1 элементе LINQ в 1.5 раза быстрее в классическом фреймворке и быстрее в Юнити.

В любом случае, быстродействие работы с коллекциями – это тема для отдельного большого исследования.

Быстродействие работы со строками

На рисунке 5 приведены замеры быстродействия строковых операций. Здесь Str1 = «1», Str10 = «1234567890», Str100 = «1234567890123...90» – до длины в 100 символов.

Можно отметить, что операции сложения строк в целом довольно тяжеловесны, также, как и преобразования чисел в строки. Быстродействие методов String.Format уступает по скорости интерполяции строк примерно на 15 %, а интерполяция, в свою очередь, уступает конкатенации примерно на столько же. String.Join, естественно существенно опережает обе из перечисленных функций.

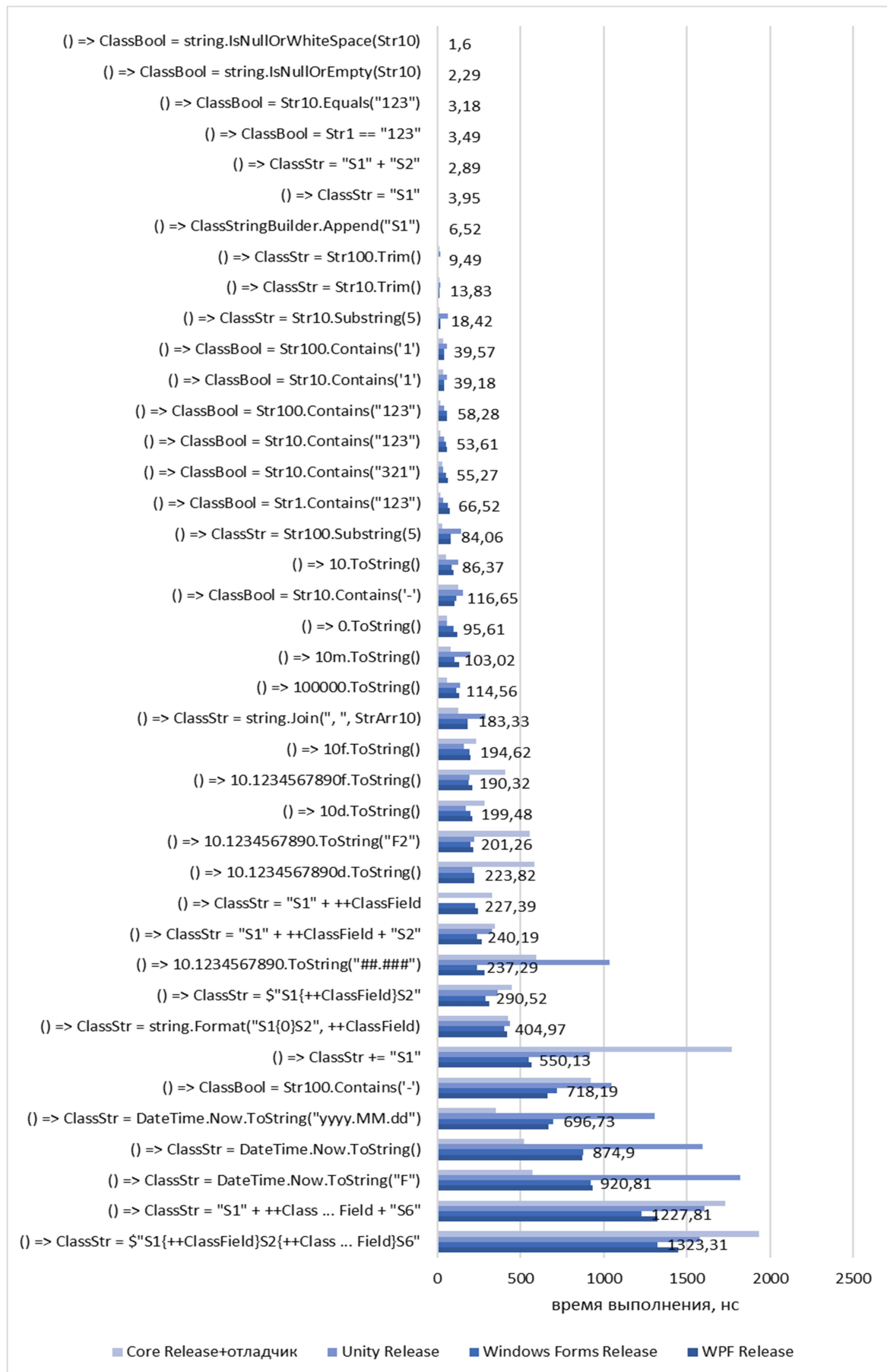


Рисунок 5. Быстродействие работы со строками в различных окружениях

Выводы

При сравнении результатов быстродействия WPF и WindowsForms в релиз-режиме получено, что средняя разница быстродействия операций по разным группам составляет до 10 %, что может быть обусловлено погрешностями измерений. В целом же все операции выполняются примерно с одинаковой скоростью, что не удивительно, ввиду единого фреймворка. Это довольно очевидно и без исследования и приведено в большей степени для того, чтобы можно было со стороны оценить погрешность измерения.

Что касается сравнения с Юнити и .Net Core, то фреймворк уже отличается, ввиду чего и отличия более существенные.

Вызовы пустых функций и обращения к переменным в Юнити выполняются, в среднем, на 43 % медленнее (за счет обращения к свойствам), математические вычисления – на 25 % (за счет таких функций, как \sin , \cos , \tan , $\sqrt{}$, abs – медленнее в 10-20 раз, тогда как atan , random , row выполнялись быстрее в 1,5–2 раза), работа с коллекциями – на 25 %, а работа со строками на 10 % медленнее. Первоначальные замеры показывали, что работа со строками в Юнити происходит медленнее, чем в WPF, на 98 % по формуле 1 (в 100 раз), но, после минимизации вызовов сборщика мусора, этот результат существенно улучшился. Тем не менее, при относительно долгом функционировании, сборка мусора в любом случае внесет свой вклад в быстродействие реальной программы.

В .Net Core базовые операции, в среднем, на 43 % медленнее, математические имеют довольно большой разброс и выполняются на 17 % быстрее (до 20 раз быстрее для atan , row и до 4 раз медленнее для \log), коллекции работают немного медленнее с малыми объемами и немного быстрее с большими – в среднем одинаково. Что же касается строк, то здесь среднее быстродействие то же, что и в WPF. Однако быстродействие выполнения отдельного функционала различается до 2–3 раз (наибольшие различия: $\text{IsNullOrWhiteSpace}$ в WPF быстрее в 6 раз, а Contains в 3 раза медленнее).

Таким образом можно заметить, что при близких результатах измерений по большинству рассмотренных операций, даже в родственных средах разработки в отдельных случаях есть принципиальные различия быстродействия часто используемых операций, которые имеет смысл учитывать при написании ПО.

Список литературы

1. Моё разочарование в софте [Электронный ресурс] // Хабр. Режим доступа: <https://habr.com/post/423889/>
2. Публичный репозиторий с кодом проекта [Электронный ресурс]. Режим доступа: <https://bitbucket.org/Konstatos/timetest/src/master/readme.md>
3. Четверина О. А. Повышение производительности кода при однофазной компиляции // Программирование. 2016. № 1. С. 51–59.

List of references

1. My disappointment in software, Habr. Mode of access: <https://habr.com/post/423889>.
2. Public repository with project code. Mode of access: <https://bitbucket.org/Konstatos/timetest/src/master/readme.md>.
3. Chetverina, O. A., “Improved code performance with single phase compilation”, *Programming*, 2016, no. 1, pp. 51–59.

МАКРИДЕНКО Е. Л., ЭМЕКСУЗЯН А. Р.
РАЗВИТИЕ ТЕХНОЛОГИИ БЛОКЧЕЙН В ЦИФРОВОЙ ЭКОНОМИКЕ
РОССИИ

УДК 338.2, ВАК 08.00.05, ГРНТИ 06.81.12

Развитие технологии блокчейн
в цифровой экономике России

The development of blockchain
technology in the digital economy of
Russia

Е. Л. Макриденко, А. Р. Эмексузян

E. L. Makridenko, A. R. Emeksuzyan

Московский политехнический
университет, г. Москва,
Коми Республиканская академия
Государственной Службы и
Управления, г. Сыктывкар

Moscow Polytechnic University,
Moscow,
Komi Republican Academy of Public
Service and Management, Syktyvkar

В данной научной статье представлены перспективы развития цифровых технологий и механизм их внедрения с использованием методологии системного анализа в государственную программу «Цифровая экономика Российской Федерации» позволяющей повысить эффективность национальной безопасности страны.

This scientific article presents the prospects for the development of digital technologies and the mechanism for their implementation using the methodology of system analysis in the state program "Digital Economy of the Russian Federation" that allows to increase the effectiveness of the national security of the country.

Ключевые слова: блокчейн, цифровые технологии, биткойн, базы данных, система майнинг.

Keywords: blockchain, digital technologies, bitcoin, databases, system mining.

На сегодняшний день развитие цифровой экономики на основе использования цифровых технологий стало ведущим мировым трендом. Становится общепризнанным, что цифровая экономика является ключевым фактором роста и развития. Для создания необходимых условий по развитию цифровой экономики в 2017 г. в России была принята программа «Цифровая экономика Российской Федерации». Как отметил Президент России В. В. Путин, от того, насколько будет развита цифровая экономика, зависит национальная безопасность и независимость страны, конкурентоспособность компаний, позиции страны на мировой арене на долгосрочную перспективу [1].

Данная программа носит комплексный и долговременный характер. К основным задачам следует отнести:

- разработка и широкое использование новых технологий;
- большие данные;

- нейротехнологии и искусственный интеллект;
- распределенные реестры;
- машинное обучение;
- виртуальная и дополненная реальность;
- квантовые технологии и т. д.

Одной из самых востребованных технологий в финансово-кредитной сфере наряду с искусственным интеллектом, стала система распределенных реестров – блокчейн (blockchain). Что такое блокчейн и для чего он нужен? Технически блокчейн – это распределенная по всему миру и управляемая автономно база данных. Устройства их хранения и обработки не замкнуты на какой-то единый сервер. База данных состоит из архивов упорядоченных записей (блоков), каждый из которых содержит в себе кодированный номер и так называемый hash, соединяющий пазлы цепочки между собой.

Блоки можно сравнить с файлами реестра вашего компьютера, с той разницей, что эти файлы хранятся одновременно на миллионах компьютерах и являются публичными. В алгоритме применяется шифрование, которое решает две задачи:

1. гарантирует безопасность данных пользователя, который владеет закрытым ключом, подтверждающим подлинность цифровой подписи;
2. обеспечивает синхронизацию копий распределенной цепочки блоков у пользователей по всему миру.

Общедоступность и одновременно 100 % безопасность блокчейн обеспечивается:

- сложными математическими алгоритмами;
- специальными программами криптографирования;
- пятью тысячами мощных компьютеров, включенных в систему майнинга, между которыми распределена вся совокупность данных.

Взломать такую систему теоретически возможно, зато практически – совершенно бессмысленно, так как никакой доход заведомо не покроет огромных расходов на глобальную атаку. Таким образом, технология блокчейн делает возможным хранение данных о финансовых операциях, юридических обязательствах, правах собственности, обеспечивая полную прозрачность и всеобщую доступность для ознакомления, но при этом надежно защищая от любого подлога, взлома и так далее. В еще более простом варианте можно сказать, что технология блокчейн – это некий стеклянный куб с постоянно включенной камерой наблюдения – в него можно (под присмотром) положить что-то новое, но при попытке изменения или подмены содержимого это тут же станет видно любому наблюдателю.

В развитии технологии блокчейн можно выделить три этапа.

Первый этап – это блокчейн сети Биткойн. Данная технология характерна только для осуществления переводов цифровой валюты. Что же такое биткойн? Это новое поколение децентрализованной цифровой валюты, созданной и работающей только в сети интернет. Её никто не контролирует, эмиссия валюты происходит посредством работы миллионов компьютеров по всему миру с ис-

пользованием программы для вычисления математических алгоритмов. Именно в этом заключается суть биткойна.

Второй этап связан с появлением технологии Ethereum, которая расширила функционал блокчейна, реализовав механизм децентрализованного выполнения алгоритмов (программного кода), позволяя создавать и использовать смарт-контракты («умные контракты»). Напомним, что эти контракты представляют собой цифровой алгоритм, описывающий набор условий, выполнение которых влечет за собой некоторые события в реальном мире или цифровых системах. В отличие от традиционных соглашений «умные контракты» не только содержат информацию об обязательствах сторон и штрафах за их нарушение, но и сами автоматически обеспечивают выполнение всех условий договора. На этом этапе технология блокчейн развилась в многовариантную платформу для различных направлений ее применения.

И наконец, **третий, актуальный этап** характеризуется разработкой обобщающих блокчейн-платформ (general purpose blockchain solutions), направленных на совершенствование и расширение возможностей протоколов смарт-контрактов, позволяющих строить масштабируемые децентрализованные приложения в очень широком спектре бизнес-процессов [2]. Развитие блокчейн в России в настоящее время тормозится противоречивым отношением к технологии со стороны разных ведомств. Тем более, Минфин РФ предлагал ввести уголовную ответственность за использование криптовалют («суррогатных денег»), а в Следственном комитете инициативу поддержали.

В то же время, с публичной поддержкой блокчейн выступил глава Сбербанка Герман Греф, а глава ЦБ РФ Эльвира Набиуллина справедливо призвала разделять понятия технологии блокчейн и цифровых валют. По мнению ряда экспертов, использование блокчейн способно помочь стране в преодолении последствий западных санкций. Например, в случае отключения от все той же системы межбанковских платежей SWIFT российские банки могут оперативно перейти на другую платформу, созданную с технологией блокчейн. Платежная система Qiwi даже готовится вводить в обращение новую российскую криптовалюту – битрубль. Впрочем, как легализация битрубля, так и использование в любой сфере «распределенного реестра» вместо «централизованного» требует основательных изменений законодательства [3].

Наряду со Сбербанком активно участвуют в разработке и применении блокчейна АЛЬФА-БАНК и Райффайзенбанк. Так, АЛЬФА-БАНК совместно с ООО «Сбербанк Факторинг» и крупнейшим в России ритейлером ПАО «М Видео» участвовал в разработке открытой платформы для применения блокчейн-технологий в области финансирования поставщиков на условиях отсрочки платежа (факторинга). Райффайзенбанк принял участие в проекте Национального расчетного депозитария по выпуску облигаций ПАО «МегаФон» с использованием технологии blockchain [2].

Таким образом, реальное применение технологии блокчейн показывает ее значимость для банковских операций. Сокращение затрат, экономия времени и безопасность транзакций – вот что привлекает в этой технологии. Задача состо-

ит в том, чтобы от экспериментального этапа перейти к промышленному использованию.

Список литературы

1. Выступление Президента России В. В. Путина на заседании Совета по стратегическому развитию и приоритетным проектам 5 июля 2017 г. [Электронный ресурс]. Режим доступа: <http://kremlin.ru/events/president/news/54983> (дата обращения 01.12.2017).

2. «Деловая среда» запустила сервис для заключения сделок на блокчейне [Электронный ресурс]. Режим доступа: <https://forklog.com/delovaya-sredazapustilaservis-dlya-zaklyucheniya-sdelok-na-blokchejne/> (дата обращения – 03.12.2017).

3. Блокчейн простыми словами [Электронный ресурс]. Режим доступа: <http://investprofit.info/blockchain>.

List of references

1. Speech by Russian President Vladimir Putin at the meeting of the Council for strategic development and priority projects on July 5, 2017. Mode of access: <http://kremlin.ru/events/president/news/54983>, accessed December, 01, 2017.

2. Business environment has launched a service for transactions on the blockchain, access mode: <https://forklog.com/delovaya-sredazapustilaservis-dlya-zaklyucheniya-sdelok-na-blokchejne/>, accessed December, 03, 2017.

3. Blockchain in simple words, access mode: <http://investprofit.info/blockchain/>

СЕМЕРИКОВ А. В.
ОЦЕНКА СОГЛАСОВАННОСТИ МНЕНИЙ СТУДЕНТОВ
 УДК 378.141.21:330.47, ВАК 05.13.01:08.00.05

Оценка согласованности мнений
студентов

Assessment of consistency
of students opinions

А. В. Семериков

A. V. Semerikov

Ухтинский государственный
технический университет, г. Ухта

Ukhta State
Technical University, Ukhta

В статье рассматривается количественный показатель оценки важности и полезности дисциплин, предложенных в процессе обучения студентов в университете. Для получения оценки использован статистический метод. В качестве критерия был принят коэффициент конкордации Кендела и коэффициент ранговой корреляции Спирмена. Проведенные исследования позволили установить согласованность мнений студентов на первом и четвертом курсах. В то же время согласованность мнений на втором и третьем курсах находится на низком уровне.

The article deals with the quantitative indicator of assessing the importance and usefulness of the disciplines proposed in the process of teaching students at the University. A statistical method was used to obtain an estimate. The criterion was adopted by the Kendal coefficient of concordance and Spearman's rank correlation. The conducted researches allowed to establish consistency of opinions of students at the first and fourth courses. At the same time, the consistency of opinions in the second and third courses is low.

Ключевые слова: статистический метод, коэффициент конкордации, коэффициент ранговой корреляции, критерий согласования Пирсона, согласованность мнений экспертов.

Keywords: statistical method, the coefficient of concordance the coefficient of rank correlation, the criterion of matching Pearson the consistency of experts opinions.

Введение

На протяжении учебы в университете студентам предлагаются для освоения множество дисциплин. Перечень дисциплин определен государственным образовательным стандартом. В результате освоения всех дисциплин студент получает соответствующее квалификационное удостоверение. При поступлении в университет студент сам выбирает интересующую его специальность на

основании имеющейся информации. В то же время, как правило, он не знаком с перечнем и содержанием изучаемых дисциплин. Поэтому в ходе обучения студент может испытать любовь к одним предметам или разочарование к другим.

Образовательный процесс представляется более эффективным, когда большинство студентов одинаково оценивают предложенные дисциплины. Под словом одинаково подразумевается большое совпадение их мнений о полезности дисциплин в дальнейшей учебной и трудовой деятельности. При этом, естественно, предпочтения студентов могут быть и диаметрально противоположные. Для определения количественного показателя мнений студентов необходимо провести формализованный сбор информации о предпочтениях студентов и выполнить обработки полученных сведений специальным способом.

Для получения числовой характеристики общего суждения студентов представляется возможным использовать статистические методы.

Экспериментальная часть

Английским статистиком Кенделом [1] был предложен коэффициент конкордации W , позволяющий оценивать степень согласованности мнений экспертов.

$$W = \frac{12 \cdot S}{M^2 \cdot (N^3 - N) - M \cdot \sum_{i=1}^M T_i},$$

где
$$T_i = \frac{1}{12} \cdot \sum_{t_j^{(i)}} \left(\left(t_j^{(i)} \right)^3 - t_j^{(i)} \right); S = \sum_{j=1}^N \left[\sum_{i=1}^M x_{ij} - \frac{1}{2} \cdot M \cdot (N + 1) \right]^2;$$

$t_j^{(i)}$ – число одинаковых рангов поставленных i -ым студентом; M – количество студентов; N – количество дисциплин.

Будучи теоретически обоснованным, он позволяет с достаточной степенью уверенности судить о согласованности мнений экспертов в некоторой группе. В настоящее время коэффициент конкордации широко используется для оценки качества продукции на основе опроса экспертов [2].

Применительно к рассматриваемой задаче в качестве экспертов выступают студенты, а в качестве группы выступает перечень дисциплин. Перед проведением расчета коэффициента конкордации W необходимо провести опрос студентов об их предпочтениях. Суть его состоит в следующем. Каждому студенту предлагается анонимно определить ранг дисциплины. Самый высокий ранг 1 получает дисциплина с наибольшим предпочтением. Самый низкий ранг получает дисциплина с самым низким предпочтением, который равен числу количества всех оцениваемых дисциплин N . Все оставшиеся дисциплины могут получить ранги равными числам между 1 и N . Если студент затрудняется отдать предпочтение одному предмету перед другими, то он может приписывать двум или более дисциплинам одинаковые ранги. Все полученные результаты опроса записываются в таблицу.

Оценки значимости коэффициента конкордации вычисляется по критерию согласия Пирсона χ^2 :

$$\chi^2 = \frac{12 \cdot S}{M \cdot N \cdot (N + 1) - \frac{1}{N - 1} \cdot \sum_{i=1}^M T_i}.$$

Для определения согласованности мнений и динамики предпочтений были проведены анонимные опросы студентов с первого по четвертый курсы.

На первом курсе в первом семестре студентам по направлению подготовки 09.03.02 Информационные системы и технологии и направлению подготовки были предложены следующие 8 дисциплин:

1. Математика.
2. История.
3. Физика.
4. Теория алгоритмов.
5. Информатика.
6. Робототехника.
7. Физкультура.
8. Иностранный язык.

При этом информацию о робототехнике студенты по направлению подготовки 09.03.02 получали самостоятельно из открытых источников. Результаты опроса студентов представлены в таблице 1.

Таблица 1. Результаты опроса 1 курса

№ студента	Дисциплины							
	1	2	3	4	5	6	7	8
1	5	8	7	1	3	4	6	2
2	3	8	6	1	5	4	7	2
3	4	6	5	1	2	3	8	7
4	3	7	8	1	2	4	6	5
5	5	6	3	1	2	4	8	7
6	5	8	7	1	3	2	6	4
7	3	8	4	1	2	5	7	6
8	1	7	6	2	3	3	8	5
9	3	6	5	1	2	4	8	7
10	2	1	5	1	3	8	7	4
11	4	8	6	1	2	5	7	3
12	5	6	7	1	2	3	8	4

Согласно представленным данным коэффициент конкордации $W = 0,7$, расчетный коэффициент $\chi^2 = 58,8$, который больше табличного $\chi^2 = 14,06$ для степеней свободы $8 - 1 = 7$ и уровне значимости $0,05$. Поэтому можно утверждать, что величина коэффициента конкордации не случайная, согласованность действительно высокая.

Для иллюстрации динамики предпочтений далее представлены результаты расчета коэффициента конкордации для студентов второго курса. При этом был проведен опрос мнений среди учащихся текущего второго курса как за предыдущий первый курс, так и за текущий второй курс (табл. 2 и 3).

Таблица 2. Результаты опроса 2 курса по дисциплинам 1 курса

№ студента	Дисциплины							
	1	2	3	4	5	6	7	8
1	2	8	3	1	1	1	5	5
2	8	7	1	1	2	1	6	2
3	5	5	3	1	1	9	3	1
4	4	5	1	8	3	6	7	2
5	4	5	7	1	8	6	3	2
6	1	6	2	8	7	5	1	3
7	2	5	3	7	1	4	6	8
8	4	3	3	5	5	2	5	1
9	1	8	6	1	4	3	5	2

Согласно представленным данным коэффициент конкордации $W = 0,2$. Расчетный $\chi^2 = 13,2$ больше табличного $\chi^2 = 12,0$ для степени свободы $8 - 1 = 7$ и уровне значимости $0,1$. Поэтому можно утверждать, что величина коэффициента конкордации не случайная, согласованность действительно низкая. В то же время для уровня значимости $0,05$ величина коэффициента конкордации случайная. В этом случае для более обоснованного принятия решения необходимо проведение дополнительных исследований.

Таблица 3. Результаты опроса 2 курса по дисциплинам 2 курса

№ студента	Дисциплины								
	1	2	3	4	5	6	7	8	9
1	3	3	8	5	1	1	2	9	7
2	4	3	1	1	1	1	2	9	4
3	5	4	1	6	1	1	4	7	7
4	1	2	4	1	1	1	2	1	5
5	5	9	8	6	1	9	3	7	2
6	4	3	6	7	1	2	5	9	8
7	7	6	5	1	4	8	9	3	2
8	2	6	8	3	4	5	9	7	1
9	1	2	5	2	3	1	1	1	5
10	3	2	4	2	1	9	2	9	1

Согласно представленным данным коэффициент конкордации $W = 0,35$. Расчетный $\chi^2 = 27$ больше табличного $\chi^2 = 15,5$ для степени свободы $9 - 1 = 8$ и уровне значимости $0,05$. Поэтому можно утверждать, что величина коэффициента конкордации не случайная, согласованность действительно низкая.

Подобным образом были проведены расчеты коэффициента конкордации для всех четырёх курсов. Результаты расчетов представлены в таблице 4.

Таблица 4. Коэффициенты конкордации с 1-го по 4-ый курсы

Эксперты (курсы)	<i>W</i> за <i>i</i> -ый год обучения			
	1	2	3	4
1	0,7			
2	0,2	0,35		
3	0,28	0,2	0,39	
4	1	0,68	1	0,54

На основании данных таблицы 4 можно утверждать следующее. Уровень согласованности мнений о предпочтениях студентов на первом и четвертом курсах высокий. Противоположные мнения наблюдаются у студентов второго и третьего курсов. Можно предположить несколько причин такого рассогласования мнений. Первая причина может быть обусловлена ошибочным заполнением предпочтений. Второй причиной такой рассогласованности может быть разделение студентов на несколько групп по предпочтениям. Для выявления этих групп можно использовать ранговый коэффициент корреляции Спирмена или Кендалла.

Для анализа полученных результатов опроса студентов и установления связи мнений студентов был использован ранговый коэффициент корреляции Спирмена. Для исследования были выбраны суждения студентов второго и третьего курсов, так как здесь наблюдается большая несогласованность мнений при оценке ранее читаемых дисциплин. В тоже время, следует заметить, по этим же дисциплинам между студентами первого курса наблюдается хорошая согласованность мнений. Для установления групп студентов с совпадающими мнениями были проведены парные сравнения опросов, представленные в таблицах 2 и 3. Полученные коэффициенты корреляции $R_{i,j}$ представлены в таблицах 5 и 6, где i и j номера студентов.

Таблица 5. Ранговый коэффициент корреляции Спирмена $R_{i,j}$ по результатам опроса 2 курса по дисциплинам 1 курса

n/R	Номера попарно сравниваемых студентов, коэффициенты ранговой корреляции						
$i-j$	1-2	1-3	1-4	1-5	1-6	1-7	1-8
$R_{i,j}$	0,57	0,14	0,1	0,31	0,43	0,17	0,52
$i-j$	2-1	2-3	2-4	2-5	2-6	2-7	2-8
$R_{i,j}$		0,07	0,12	0,05	0,26	0,19	0,24
$i-j$	3-1	3-2	3-4	3-5	3-6	3-7	3-8
$R_{i,j}$			0,21	0,21	0,38	0,2	0,17
$i-j$	4-1	4-2	4-3	4-5	4-6	4-7	4-8
$R_{i,j}$				-0,5	0,36	0,52	-0,29
$i-j$	5-1	5-2	5-3	5-4	5-6	5-7	5-8
$R_{i,j}$					-0,83	-0,12	0,57
$i-j$	6-1	6-2	6-3	6-4	6-5	6-7	6-8
$R_{i,j}$						-0,19	0,21
$i-j$	7-1	7-2	7-3	7-4	7-5	7-6	7-8
$R_{i,j}$							-0,21

Представленные в таблице 4 результаты расчета рангового коэффициента корреляции позволяют принять нулевую гипотезу о нулевой зависимости суждений при парном сравнении, так как для уровня значимости $p = 0,05$ критическое значение коэффициента корреляции Спирмена $R_{крит} = 0,72$, что существенно больше представленных выше коэффициентов.

Такой же расчет коэффициента Спирмена проведен по данным опроса студентов второго курса по дисциплинам второго курса. Результаты расчета представлены в таблице 6.

Таблица 6. Ранговый коэффициент корреляции Спирмена $R_{i,j}$ по результатам опроса 2 курса по дисциплинам 2 курса

n/R	Номера попарно сравниваемых студентов, коэффициенты ранговой корреляции							
$i-j$	1-2	1-3	1-4	1-5	1-6	1-7	1-8	1-9
$R_{i,j}$	0,65	0,75	0,23	0,07	0,88	-0,6	0,02	0,15
$i-j$	2-1	2-3	2-4	2-5	2-6	2-7	2-8	2-9
$R_{i,j}$		0,8	0,18	-0,05	0,6	0,13	-0,07	-0,23
$i-j$	3-1	3-2	3-4	3-5	3-6	3-7	3-8	3-9
$R_{i,j}$			0,08	-0,13	0,83	-0,6	-0,3	-0,18
$i-j$	4-1	4-2	4-3	4-5	4-6	4-7	4-8	4-8
$R_{i,j}$				-0,08	0,17	-0,02	-0,08	0,68
$i-j$	5-1	5-2	5-3	5-4	5-6	5-7	5-8	5-9
$R_{i,j}$					-0,05	0,23	0,38	-0,13
$i-j$	6-1	6-2	6-3	6-4	6-5	6-7	6-8	6-9
$R_{i,j}$						-0,57	0	-0,1
$i-j$	7-1	7-2	7-3	7-4	7-5	7-6	7-8	7-9
$R_{i,j}$							0,47	-0,37
$i-j$	8-1	8-2	8-3	8-4	8-5	8-6	8-7	8-9
$R_{i,j}$								-0,3

Представленные в таблицах 5, 6 результаты расчета коэффициента корреляции парных сравнений свидетельствуют об отсутствии общего мнения студентов о полезности читаемых дисциплин. При этом не представляется возможным выделить группы студентов с явно выделенными предпочтениями.

Для выявления второй причины такого рассогласования был проведен повторный опрос студентов по прошествии четырех месяцев. Результаты повторного опроса представлены в таблице 7.

Таблица 7. Результаты повторного опроса студентов 2 курса по дисциплинам 1 курса

№ студента	Номера дисциплин							
	1	2	3	4	5	6	7	8
1	6	1	3	8	8	7	8	5
2	8	6	8	5	7	4	7	8
3	2	5	4	7	7	1	7	6
4	1	5	1	3	2	1	1	5
5	8	4	5	8	8	7	2	8

6	7	1	3	8	6	7	5	7
7	7	4	4	4	8	8	8	8
8	7	4	4	6	8	8	7	8
9	7	4	4	6	8	8	7	8
10	6	1	3	8	8	7	8	5

Коэффициент конкордации в этом случае равняется $W=0,29$. Этот результат вновь свидетельствует о смене общих предпочтений о полезности читаемых дисциплин.

Выводы

На основании проведенных исследований представляется возможным утверждать следующее:

1. Уровень согласованности мнений о предпочтениях студентов на первом и четвертом курсах высокий. Противоположные мнения наблюдаются у студентов второго и третьего курсов.

2. При высоком уровне согласованности явно просматриваются как лидеры (дисциплины), так и аутсайдеры в предпочтениях. При этом не представляется возможным оценить этот факт. С одной стороны, это положительное явление, а с другой стороны это можно трактовать как отрицательный результат. При высоком согласии большинство студентов оказывают предпочтение одним и тем же предметам, тем самым наблюдается перекося в одну сторону. При низкой согласованности наблюдается широта интересов.

3. Предпочтения студентов не постоянны во времени. Так, высокие предпочтения к предметам на первом курсе уменьшаются по прошествии одного и более лет. Тем самым отмечается переосмысление мнений по одним и тем же предметам.

4. Для более точной оценки мнений студентов необходимо проведение повторных опросов с различным составом студентов, чтобы установить наличие групп студентов, оказывающих большое влияние на уровень согласованности мнений. Тем самым можно установить, меняются ли предпочтения студентов через год и более. И в конечном счете результаты расчетов можно будет использовать для корректировки учебного процесса.

Список литературы

1. Kendall, M. G.; Babington Smith, B. (Sep 1939). The Problem of m Rankings. *The Annals of Mathematical Statistics* 10 (3): 275–287.

2. Основы экономического прогнозирования [Электронный ресурс]. Режим доступа: <http://bibook.ru/books/32686/default.htm>.

List of references

1. Kendall, M. G., Babington Smith, B. (Sep 1939) "The Problem of m Rankings", *The Annals of Mathematical Statistics*, 10 (3): 275–287.

2. Basics of Economic Forecasting. Mode of access: <http://bibook.ru/books/32686/default.htm>.

МАКРИДЕНКО Е. Л., ЭМЕКСУЗЯН А. Р.
ПРОГНОЗИРОВАНИЕ ИЗМЕНЕНИЙ ДЛИТЕЛЬНОСТИ ЭТАПОВ
ЖИЗНЕННОГО ЦИКЛА ИННОВАЦИОННОГО ПРОДУКТА
УДК 52(15):33, ВАК 08.00.05, ГРНТИ 89.01.75

Прогнозирование изменений
длительности этапов жизненного цикла
инновационного продукта

Е. Л. Макриденко, А. Р. Эмексузян

Московский политехнический
университет, г. Москва,
Коми Республиканская академия
Государственной Службы и Управления,
г. Сыктывкар

В данной научной статье проведен анализ системной оценки длительности этапов жизненного цикла инновационного продукта и представлены перспективы прогнозирования и внедрения с использованием методологии системного анализа в государственную программу опережающего развития экономики страны, позволяющей повысить эффективность национальной безопасности страны.

Ключевые слова: инновации, инновационный продукт, жизненный цикл, прогнозирование, модель.

Prediction of changes in the duration
of the stages of the life cycle of an
innovative product

E. L. Makridenko,
A. R. Emeksuzyan

Moscow Polytechnic University,
Moscow,
Komi Republican Academy of
Public Service and Management,
Syktyvkar

This scientific article analyzes the systemic assessment of the duration of the life cycle stages of an innovative product and presents the prospects for forecasting and implementation using the methodology of system analysis in the state program of advanced development of the national economy, allowing to increase the effectiveness of the national security of the country.

Keywords: innovation, product innovation, life cycle, forecasting, model.

Одним из самых слабых условий освоения промышленных инноваций является низкий уровень их экономического обоснования. Это связано не только с недостаточным количеством специалистов требуемой квалификации и отсутствием опыта подобной работы, но и с определенным типом организационной культуры большинства научно-исследовательских организаций и организаций-заказчиков, которой присуща сильная советская «инерция», что приводит к спонтанному (и нередко волюнтаристскому) инициированию нововведений.

Одним из направлений решения проблемы для отечественных предприятий является своевременная реализация инноваций разного уровня новизны. Как справедливо отмечают американские специалисты «критическими вопросами является качество, количество и время появления новых изделий. Не менее важны – скорость и гибкость процесса». Значительная часть успеха будущей инновации кроется именно в правильном определении времени начала ее внедрения, поскольку слишком ранний выход на рынок, который еще не готов к приятию нового продукта, может обернуться значительными финансовыми потерями. Так, неправильно оценив момент времени, предприятия могут отказаться от «старой» технологии/продукцию в период, когда она еще дает прибыль, при этом также, не доведя до коммерциализации инновационный проект по причине, например, истощения финансовых ресурсов. Но и промедление с инновациями не желательно, так как пионерные разработки конкурентов, завоевывая признание потребителей, монополизировать рынок лишая тем самым своих конкурентов долей рынка.

В свою очередь, недостаточно быстрое наращивание темпов совершенствования продукции (технологий, услуг) в начале кривой может привести к прекращению финансирования или даже к досрочному отказу от новой технологии. И, наоборот, может потребоваться дополнительное инвестирование в виду завышенных предварительных оценок возможных темпов разработки новой продукции, или в виду неспособности учесть усилия других участников инновационного прогресса в отрасли, которые генерируют знания, доступные для тех, кто хочет их получить.

Наиболее распространенным подходом для определения стадии жизненного цикла, как основы определения необходимости в инновации, является модель учета технологических укладов (определенного порядка «составляющих производств, связанных вертикальными и горизонтальными потоками качественно однородных ресурсов, которые опираются на совокупные ресурсы квалифицированной рабочей силы, общий научно-технический потенциал и определенные технологии»), которая, в свою очередь, в значительной мере опирается на теории синергетики, кризисов и катастроф.

Сам технологический переход от одной технологии к другой состоит из двух этапов (эволюционного и революционного) и завершается технологическим разрывом в точке бифуркации, пороговом состоянии системы, переход через который ведет к резкому качественному изменению протекающих в ней процессов и изменению самой организации.

Во время эволюционного этапа развития системы с помощью специфических механизмов подавляются сильные флуктуации системы, ее компонентов и среды и возвращают ее в устойчивое состояние. Постепенно в системе возрастает энтропия, поскольку через накопление в системе, а также в ее компонентах и внешней среде изменений способность системы к адаптации падает и нарастает неустойчивость. Возникает острое противоречие между старым и новым в системе, а при достижении параметрами системы и среды бифуркационных значений неустойчивость становится максимальной и даже малые флуктуации приводят систему к катастрофе, скачка или кризиса.

На фазе скачка развитие приобретает непредусмотренный характер, поскольку оно вызывается не только внутренними флуктуациями, силу и направленность которых можно прогнозировать, но и внешними, что крайне усложняет, а то и делает невозможным прогноз.

Таким образом, основной задачей анализа необходимости реализации инноваций является определение критических уровней параметров системы, которые оказывают непосредственное влияние на ее состояние, и оценка устойчивости системы к изменениям. Причем главное здесь заключается в определении критического уровня неинертных параметров, которые и должны выступать в качестве индикаторов необходимости в появлении инноваций.

Анализ литературных источников [1–4] прямо или опосредованно показал, что существуют несколько методов определения необходимости в реализации инноваций:

- 1) графическое определение изменений технологических укладов за счет построения S-подобной кривой развития продукта или технологии;
- 2) построение математических моделей, которые описывают инновационный процесс, и разработка на их основе прогноза изменений фаз жизненных циклов продуктов или технологий;
- 3) определение реперных точек, значения которых обуславливают стабильность состояния предприятия, в т. ч. и с помощью матричного маркетингового аппарата.

Графический метод определения технологических разрывов заключается в следующем:

- 1) сначала определяют, на что должна быть направлена модель (класс, тип товаров, марка, конкретная модель) и конкретизируют рынок, поскольку могут быть разные фазы цикла на отечественном и зарубежном рынках;
- 2) на втором шаге определяется комплекс факторов, которые формируют процесс циклического развития производства. К таким факторам относятся: параметры S-кривой насыщения потребности в данном виде продукта, технико-технологические и экономические параметры продукции, технико-экономический уровень производства, параметры модели выпуска продукта. Далее, необходимо определить взаимосвязь и взаимообусловленность анализируемых процессов. При чем эти первые два шага используются во всех существующих методах определения времени внедрения инновации;

3) строят S-подобную кривую, которая выражает зависимость между эффектом от деятельности организации (ось ординат) и затратами (ось абсцисс) и путем построения проекций точек изменения направления кривой определяют бифуркационное множество, которое содержит те значения параметров, при которых возможны несколько разных состояний системы. Американские специалисты утверждают, что, когда крутая кривая начинает выравниваться, пришло время изменить направление усилий по совершенствованию продукта или процесса, обращая внимание на другие параметры.

Основным недостатком, сдерживающим эффективное использование данного метода, является то, что в современной практике усилия по разработке инноваций измеряют затраченным на их реализацию временем, поскольку разные

компании начинают и прекращают программы НИР, а также финансируют их на разных этапах. При этом в качестве объекта измерения выступают лишь собственные усилия, тогда как на развитие любого объекта за счет направления усилий контрагентов на развитие и модернизацию такой же продукции/технологии или продукции-комплимента влияет также и внешняя среда. И вдобавок большинство компаний не ведет учета своих усилий, которые были вложены в определенную технологию, и преимущественно стараются построить кривую жизненного цикла предыдущей инновации относительно времени, в связи с чем оказывается, что прогнозы часто не оправдываются.

Проблема здесь заключается не в трудностях прогнозирования технического прогресса, поскольку, как утверждает, например, Р. Фостер S-кривые развития относительно стабильны, а, скорее, в неумении отслеживать и прогнозировать инвестиции всех наиболее крупных производителей конкретной отрасли. Чтобы построить семейство S-кривых, необходимо реконструировать и прогнозировать усилия основных участников данной отрасли, исходя из их расходов на НИР, или из более прямого критерия, которым является количество лет, потраченных на разработку определенной технологии и ее заменителей, что весьма сложно осуществимо в условиях рыночной конкуренции, нежеланием предприятий раскрывать основу своих конкурентных позиций. Кроме того, абсолютно не учитывается ни нежелание системы в целом изменяться, ни предельный уровень чувствительности технологии к изменениям, достижение которого предшествует появлению нового товара, технологии или услуги, что и обуславливает ограниченность использования графического метода.

Другой группой методов прогноза изменения этапов жизненного цикла продукта, как основы определения необходимости в инновациях, является построение математических моделей изменения состояния как внутренней, так и внешней среды функционирования предприятия. Большинство из этих моделей основаны на использовании гипотезы, которую выдвинул Я. ван Дейн, согласно которой появление инноваций разных типов связано с разными фазами социально-экономического и научно-технического развития, представленными в виде «длинной волны», и на теории устойчивости Флоке-Ляпунова.

Так, в частности Б. Федешиним смоделирован процесс обновления основных производственных фондов (ОПФ) и насыщение рынка новыми видами продукции. В основу разработанной методики положена функция. Перспективу (Y) обеспечение новыми производственными фондами, прогноз внедрения новых технологических процессов, освоение производства новых видов продукции можно определить по формуле:

$$Y = \frac{C_1 \cdot A}{C_1 + e^{-bt}},$$

где A – насыщение новыми производственными фондами, видами продукции; t – время;

$$C_1 = e^A, b = k,$$

где k – коэффициент пропорциональности; e – постоянная интегрирования.

Несмотря даже на отсутствие раскрытия содержания всех составных элементов вышеуказанной формулы, вызывает сомнение возможность ее практического использования в связи со сложностью определения уровня насыщения производственными фондами и продукцией даже в пределах небольшого региона. Такое состояние дел обусловлено отсутствием статистических данных как касающиеся начального уровня ОПФ и изготавливаемой предприятием продукции, так и темпов ее замещения. В связи с этим, прогнозы, разработанные с помощью использования данной математической модели, не отличаются достаточной точностью и таким образом могут вводить в заблуждение лицо, которое принимает решение о внедрении новой продукции.

В свою очередь, с целью количественного прогнозирования состояния внешней среды Ю. Антонюк предлагает использовать многопараметрический прогноз, который базируется на определении коэффициентов реакции (a, b, c, d, e), предоставляющих информацию о силе реакции прогнозируемой переменной на изменения независимых составляющих модели, и проявляющихся специфично в уравнениях поведения. Эти коэффициенты можно заранее и часто довольно точно «адаптировать» к ближайшему будущему. В результате получим уравнение поведения вида:

$$\hat{y} = ax + bx_1 + cx_2^2 + dx_3^2 + ex_4^2$$

где e – коэффициент реакции прогнозируемого спроса на существующий товар, касающийся уровня цен определенного товара-заменителя (x_4).

Исследование возможности моделирования инновационных процессов, а значит и продолжительности их этапов, присутствуют также в работах В. Яковлева. Он предложил две модели динамического инновационного процесса, в основу которых положено подобие между изменением во времени оперативного энергopotенциала и гармонического осциллятора, с одной стороны, и экспотенциально затухающие гармонические функции, отражающие поведение оперативного энергopotенциала по модели «шагреновой кожи».

Несовершенство данной модели, как других, упомянутых выше, заключается в отсутствии механизма (алгоритма) расчета переменных, используемых в модели при одновременном отсутствии методических рекомендаций по использованию всей модели в целом.

В литературе нередко можно встретить реплику о том, что из известных науке 1380 видов экономических циклов/моделей развития товаров лишь три имеют хотя бы какое-то практическое значение. Это – циклы товарно-материальных запасов Дж. Китчина (с периодом до 3 лет), циклы инвестиций в оборудование К. Угляра (период 7–11 лет); строительные циклы С. Кузнеца (период 18–22 года). Тем не менее, и эти циклы с целью прогнозирования инновационных преобразований могут использоваться лишь для определения в лучшем случае усложняющихся инноваций.

Таким образом, отсутствие достаточно точных моделей протекания инновационных процессов (в связи с тем, что: фирмы не всегда обнаруживают ту определенную последовательность изменений во времени, как это происходит в биологических системах; вследствие большого количества факторов довольно тяжело определять продолжительность этапов жизненного цикла; отрицательным влиянием на жизненный цикл фирм и организаций ошибок их руководства) делает малоэффективным использование специального математического аппарата Флоке-Ляпунова, применяемого для анализа бифуркаций и этим самым делает явно недостаточным использование на практике методик определения необходимости в инновациях на основе большинства моделей.

Еще одним недостатком научного прогнозирования будущего развития ЖЦ продукции, в т. ч. и на основе моделей, является то, что, используя научные методы логического анализа и синтеза, предлагаемые в литературе модели выводят будущее из прошлого и настоящего (как бы экстраполируют), а на самом деле состояние среды обуславливается в значительной степени случайными процессами, свойством которых является отсутствие тесноты связей мотивационных интересов всех заинтересованных сторон. Именно это обстоятельство и обуславливает неопределенность поведения потребителя инноваций, вызванная зачастую неэластичностью цены и качества. Об этом прямо говорится в работах Й. Шумпетера, Г. Менша, Х. Фримена, Д. Сахала, М. Дибнера, М. Ловелла. Это и обуславливает невозможность создания действенных моделей случайного непоследовательного процесса, которым является реализация инноваций, поскольку отсутствуют элемент повторяемости событий, составляющий основу всякого моделирования технологически связанных звеньев производств.

Подытоживая результаты анализа первых двух групп методов определения необходимости в инновациях и времени их реализации, которые предназначены для распознавания необходимости в реализации радикальных инноваций, можно констатировать наличие в них существенных недостатков. Последние в основном связаны с очевидной ограниченностью отслеживать общие тенденции в развитии однородных продуктов или технологий в пределах всего рынка, а не только отдельно взятого пользователя.

Анализ же усовершенствующих инноваций, роль которых в экономике никак не меньше чем радикальных, отнесено к компетенции третьей группы методов – определение слабых сигналов (реперных точек) изменения состояния систем на основе маркетинговой и финансовой информации. Основу этого метода составляет принцип адаптивности структур к изменениям самой системы и ее окружение.

При этом согласно экстерналистской теории первопричины изменений любой социокультурной системы, в том числе организации, находятся за пределами самой системы. Соответственно теории имманентного изменения любая структура развивается в силу неотъемлемого свойства изменяемости на основе собственных возможностей и ресурсов. Причем внешние факторы не отрицаются, а рассматриваются как дополнение. Они могут ускорить, замедлить, остановить преобразование и даже разрушить модифицирующийся объект.

Главной идеей данных методов является то, что двигателем экономического развития и технического прогресса является та сила, которая создает предпосылки к инновационному спросу на новые виды продуктов и услуг по времени, объему и его качеству.

Список литературы

1. Абалкин Л. От экономической теории до концепции долгосрочной стратегии // Вопросы экономики. 2010. № 6.
2. Балдин К. В. Управление рисками в инновационно-инвестиционной деятельности предприятия. М. : Дашков и Ко, 2017.
3. Кузык Б. Н., Яковец Ю. В. Россия 2050: стратегия инновационного прорыва. М. : Экономика, 2015.
4. Ленчук Е. Б. Проблемы формирования инновационной системы России. М. : ИЭ РАН, 2017.
5. Новицкий Н. А. Инвестиционный потенциал развития инновационной экономики России. М. : ИЭ РАН, 2018.
6. Системы управления инновационно-инвестиционной деятельностью промышленных организаций и подготовка машиностроительного производства. Под ред. проф. А. В. Рождественского, Р. С. Голова. М. : Дашков и К, 2016.

List of references

1. Abalkin, L., "From economic theory to the concept of long-term strategy", *Economic issue*, 2010, no. 6.
2. Baldin, K. V., *Risk management in innovation and investment activities of the enterprise*, Moscow : Dashkov and Co., 2017.
3. Kuzyk, B. N., Yakovec, YU. V., *Russia 2050: strategy of innovative breakthrough*. Moscow : Economy, 2015.
4. Lenchuk, E. B., *Problems of formation of innovative system of Russia*, Moscow : Institute of economics of RAS, 2017.
5. Novickij, N. A., *Investment potential of innovative economy development in Russia*, Moscow : Institute of economics of RAS, 2018.
6. Systems of management of innovation and investment activity of industrial organizations and preparation of machine-building production, Edited by prof. A. V. Rozhdenstvenskogo, R. S. Golova, Moscow : Dashkov and Co., 2016.

ИГНАТОВ Д. А.
РАЗРАБОТКА ИГРЫ «KOREAN FOOTBALL»
 УДК 004:794.5, ВАК 05.13.18, ГРНТИ 50.41.25

Разработка игры «Korean Football»

Creation of the game “Korean Football”

Д. А. Игнатов

D. A. Ignatov

Ухтинский государственный
 технический университет, г. Ухта

Ukhta State Technical University, Ukhta

В статье рассматриваются вопросы, связанные с разработкой мобильной игры «Korean Football». Особое внимание акцентируется на исследовании предметной области, моделировании системы и описании результатов разработки программного продукта.

The article discusses the issues related to the development of the mobile game “Korean Football”. Special attention is focused on studying the domain, modeling the system and describing the results of software product development.

Ключевые слова: казуальная игра, мультиплеер, геймплей, конечный автомат

Keywords: casual game, multiplayer, gameplay, finite state machine

Введение

С каждым годом популярность компьютерных и мобильных игр все растет. Результатом технологического развития и массового распространения мобильных устройств стал все более расширяющийся рынок развлечений. Особое место занимают казуальные игры, которые в большинстве своем не требуют больших затрат средств на создание, крупного штата разработчиков, но в то же время могут быть известны не хуже всех остальных. Самые удачные проекты по созданию таких игр продаются миллионными тиражами, многие игроки готовы вкладывать в них огромные деньги. Начинающие разработчики получают исключительную возможность развить и продемонстрировать собственные навыки, сделать себе имя даже на простых играх, что позволит быть замеченными и приглашенными для участия в более крупных проектах.

Заказчиком игры, которая выступает объектом автоматизации в данной выпускной квалификационной работе, является компания по разработке игр «Unnyhog FZ-LLC». Во время пребывания в Южной Корее разработчикам было предложено реализовать настольную игру, которая в прошлом была популярна среди корейцев всех возрастов. Даже не имея прямых цифровых аналогов, игра в этой стране остается признанной и незабытой.

Создание цифрового аналога на популярных устройствах не только удовлетворит развлекательные потребности жителей Южной Кореи, которые по-

новому взглянут на модернизированную версию старой игры, но и вовлечет пользователей с других стран. Игра должна стать одним из новых свободно распространяемых продуктов компании по разработке игр «Unnyhog FZ-LLC», который поможет ей обрести известность в корейском и других рынках, собрать новых пользователей, получить необходимые связи, а также заработать деньги на внутриигровых покупках и рекламе.

Создание игры требует тщательного анализа предметной области и существующих аналогов, чтобы в результате получить программный продукт, удовлетворяющий требованиям пользователей, начиная от необходимого функционала, и заканчивая удобным интерфейсом.

Предпроектное исследование

Разрабатываемая игра берет истоки из старой настольной игры, имевшей популярность в Южной Корее. Участвуют всего два игрока. Для игры необходим только расчерченный на определенное количество полей лист (чаще всего на восемнадцать), две ручки (или любых других твердых предмета тонкой продолговатой формы) и монетка. Суть состоит в том, чтобы толкать своей ручкой (толстым концом) монетку ко вражеским воротам, роль которых исполняет крайняя средняя клетка на стороне поля соперника. Один из игроков выступает в качестве нападающего, а другой в качестве защитника. При наступлении определенных моментов в игре игроки меняются ролями.

Нападающий может ходить своей ручкой на всем пространстве поля. Количество толчков и времени хода для нападающего неограниченно. При этом ему нельзя допускать, чтобы монетка остановилась на одной из расчерченных линий или вылетала за пределы поля (листа), иначе роль нападающего перейдет сопернику. То есть ему необходимо аккуратно рассчитывать силу и направление удара, чтобы переместить монетку от одной клетки к другой, и так до ворот соперника.

При этом соперник, исполняющий роль защитника, не будет бездействовать. Он может ходить своей ручкой только по расчерченным линиям, и его задачей является выбить монетку в тот момент, когда она будет нестись через одну из линий. Если удастся, и монетка отлетит за пределы поля или на какую-либо линию, то роль нападающего перейдет защитнику, а нападающий в свое время станет защитником.

В разрабатываемом цифровом аналоге описанной настольной игры недостаточно лишь перенести эти правила, нужно их изменить для более захватывающего игрового процесса. И, конечно же, реализовать все сопутствующие большинству современных игр компоненты, такие как: дизайн, меню, базу данных, мультиплеер, искусственный интеллект при отсутствии живого соперника и т. д.

Таким образом, в перечень функциональных возможностей игры для игрока входит:

- Участие в матчах. Позволяет участвовать в основном игровом процессе в различных режимах (один против искусственного интеллекта, двое на одном устройстве, двое на разных устройствах).

– Просмотр статистики. Позволяет игрокам просмотреть свою статистику (количество побед, поражений, сдач, золота, рейтинг и другое). Статистика обновляется автоматически и незамедлительно после каждого изменения, синхронизируясь с базой данных.

– Совершение внутриигровых покупок. Позволяет заходить во внутриигровой магазин, обладающий широким выбором товаров (моделей, анимаций и других визуальных улучшений), приобретать эти товары за игровую или реальную валюту и использовать товары в дальнейшем игровом процессе.

– Авторизация и регистрация. Позволяет пользователям выполнить регистрацию и в дальнейшем авторизоваться под личным аккаунтом с целью получения доступа ко всем функциональным возможностям игры, включая уже те, что недоступны незарегистрированным игрокам.

Обзор аналогов осуществлялся с целью поиска систем, которые реализуют схожую функциональность. Одним из таких аналогов стала игра по типу аэро-хоккея Glow Hockey [1]. В результате была составлена таблица сопоставления функциональных возможностей на основе критериев:

- 1) бесплатность – отсутствие цены за приобретение программного продукта;
- 2) ИИ – наличие возможности игры против искусственного интеллекта;
- 3) мультиплеер на 1 устройстве – наличие возможности совместной игры на одном устройстве;
- 4) мультиплеер на 2 устройствах – наличие возможности совместной игры на разных устройствах;
- 5) смена внешнего вида игровых объектов – наличие возможности кастомизации внешнего вида игровых объектов в инвентаре (мяча, персонажа и т. д.);
- 6) появляющиеся на поле бонусы – наличие дополнительной особенности геймплея, позволяющей игрокам принимать больше ответственных решений, от которых будет зависеть исход матча;
- 7) квесты – наличие дополнительной особенности игры, позволяющей игрокам выполнять различные задачи помимо выигрыша в матчах и зарабатывать золото для покупки внешнего вида игровых объектов;
- 8) рейтинг и статистика – наличие отслеживаемого, фиксируемого и сравниваемого прогресса игроков.

После исследования онлайн-рынка мобильных игр было выявлено, что прямых аналогов игры с таким же геймплеем и правилами на данный момент не существует. Это и стало одной из главных причин разработки игры «Korean Football». Поэтому были рассмотрены самые близкие программные продукты.

Таблица 1. Сопоставление функциональных возможностей аналогов

	Glow Hockey	Korean Football
Бесплатность	Да	Да
ИИ	Да	Да
Мультиплеер на 1 устройстве	Да	Да
Мультиплеер на 2 устройствах	Нет	Да
Смена внешнего вида игровых объектов	Ограниченная, но бесплатная	Расширенная, покупается за выигранное в матчах золото
Появляющиеся на поле бонусы	Нет	Да
Квесты	Нет	Да
Рейтинг и статистика	Нет	Да

Проектирование информационной системы

Основываясь на анализе предметной области, была составлена контекстная диаграмма процесса «как будет» с точки зрения игры (рис. 1). Результатом декомпозиции процесса стала диаграмма потоков данных (рис. 2).

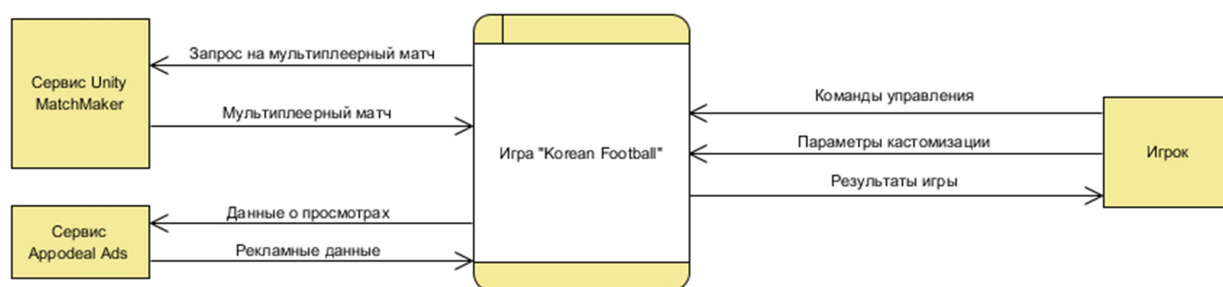


Рисунок 1. Контекстная диаграмма

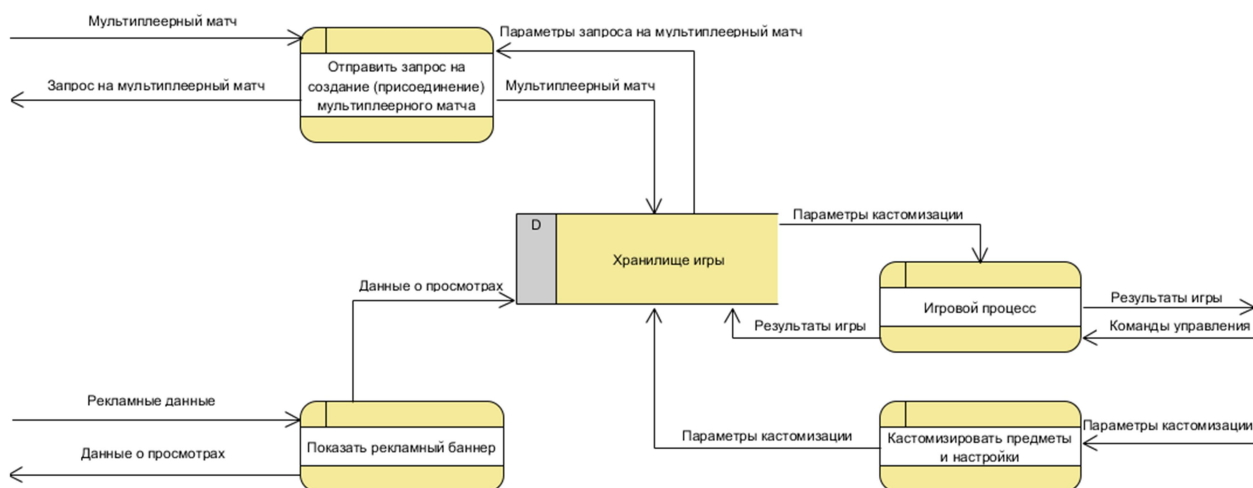


Рисунок 2. Диаграмма потоков данных

Для реализации связи с базой данных использовалась трехзвенная клиент-серверная архитектура. При такой схеме возможно централизованное администрирование прикладных функций и снижение сетевого трафика (т. к. с клиента

передаются не sql-запросы, а всего лишь формы с необходимыми параметрами, которые затем используются на серверной части) (рис. 3).

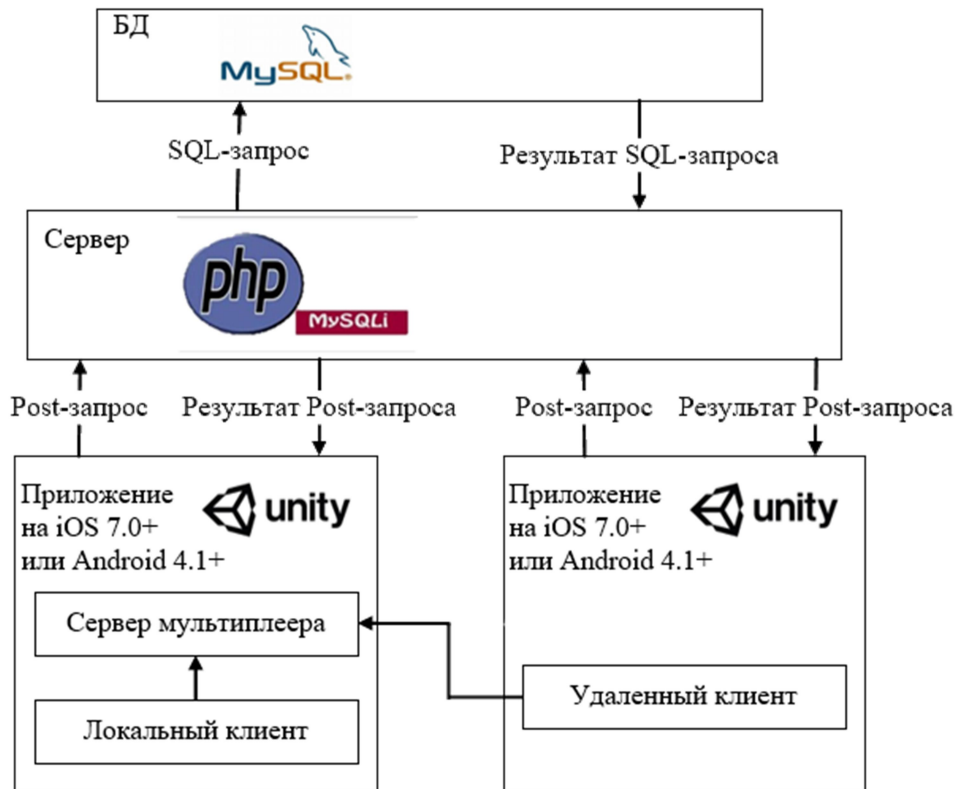


Рисунок 3. Архитектура системы

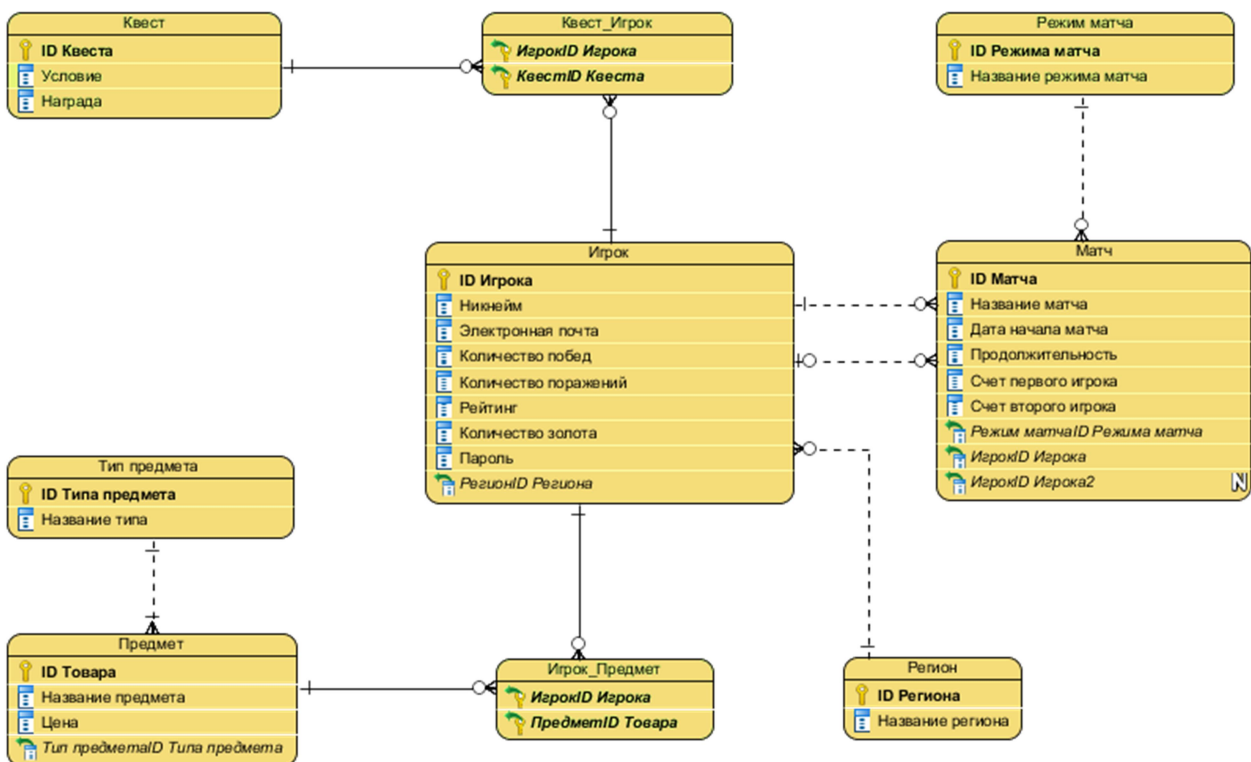


Рисунок 4. Логическая модель БД

Первым шагом при проектировании базы данных является построение логической модели, которая включает в себя перечень сущностей, выделенных в ходе описания предметной области, их атрибуты, а также взаимосвязи между сущностями (рис. 4).

Результаты разработки

Результатом разработки игры стал почти готовый к эксплуатированию программный продукт на игровом движке Unity [2], написанный на языке программирования C#. Архитектура игры представляет собой трехзвенную клиент-серверную архитектуру. Данные находятся на хостинге beget [3] и связываются с бизнес-логикой игры через php-скрипты на слое доступа к данным. Бизнес-логика основана на следующих основных контроллерах, отвечающих за определенные составляющие игры:

- контроллер меню;
- контроллер связи с базой данных;
- контроллер аудио;
- контроллер искусственного интеллекта на основе конечного автомата;
- контроллер предметов, инвентаря и магазина;
- контроллеры игрового процесса.

Одной из основных задач являлось создание пользовательского интерфейса, отражающего основные возможности продукта помимо участия в игровых матчах. Для отслеживания статистики и купленных предметов игрок может зарегистрироваться, после чего в меню игрового профиля сможет увидеть всю информацию о своем прогрессе (рис. 5).



Рисунок 5. Меню авторизации и меню профиля

На главном меню отражены самые основные функциональные кнопки, через которые пользователь сможет попасть либо сразу в матч на игровое поле,

либо к необходимому подменю. Одним из таких подменю является инвентарь, объединенный с магазином, в котором игрок сможет как купить различные вариации игровых предметов, так и надеть любой из уже купленных (рис. 6).



Рисунок 6. Главное меню и меню инвентаря

Другим подменю является меню тренировки, где пользователь может сыграть против искусственного интеллекта, против друга на этом же устройстве, или против случайного игрока через интернет (рис. 7).



Рисунок 7. Меню тренировки и меню присоединения к созданному матчу

Основной функциональной возможностью продукта, конечно же, является участие в матче. В нем можно изучить и опробовать все тонкости игрового геймплея и победить соперника (рис. 8). Модели для персонажей с их анимациями были взяты с онлайн-сервиса Mixamo [4].

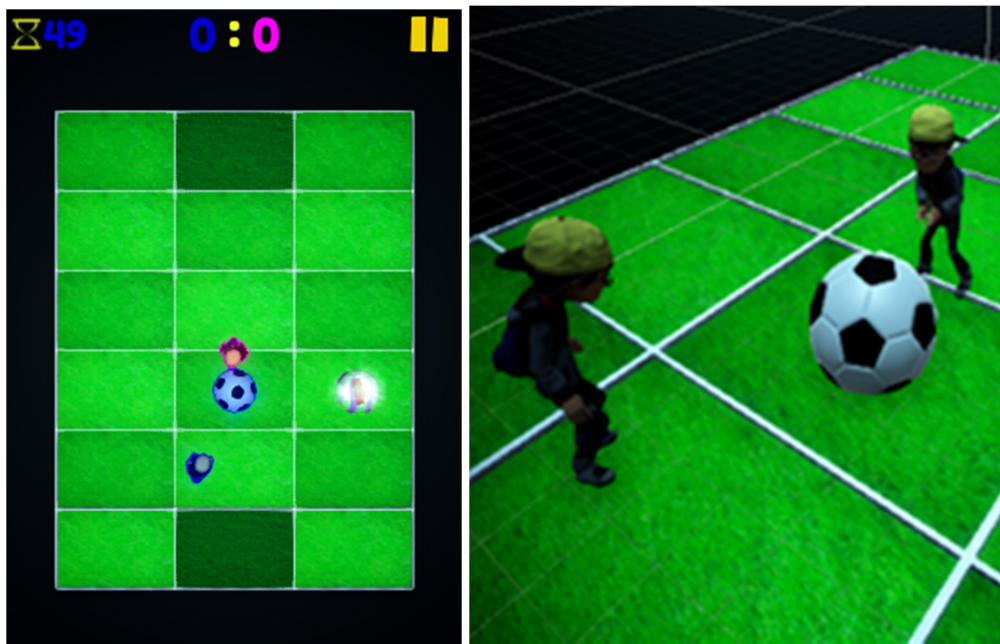


Рисунок 8. Игровая сцена с ракурса игрока и с побочного ракурса

В игре был также реализован искусственный интеллект, чтобы начинающий игрок смог сначала освоиться в игровом процессе и испытать свои навыки в режиме одиночной игры. Кроме того, если автоматический поиск соперника будет затягиваться из-за отсутствия других игроков, пользователь будет переброшен в игру против искусственного интеллекта. Именно поэтому особенно важно сделать ИИ максимально приближенным по игровому стилю к реальному игроку, чтобы любой пользователь при перебрасывании в бой с ИИ мог воспринять соперника за настоящего игрока.

Для реализации этой цели и большего разнообразия были эмулированы поведения нескольких распространенных типов игроков: «агрессора», «выжидателя» и «тормоза». В данной игре это было легко достигнуто даже путем одного лишь изменения значений констант в логике ИИ. При попадании в бой против ИИ, будет случайным образом выбран один из этих типов искусственного интеллекта со своим набором констант, а, следовательно, и со своим поведением. Само поведение основано на модели Finite State Machine – конечного автомата, в котором может быть активно только одно состояние (при котором ИИ выполняет определенный набор действий), и для выполнения другого набора действий ИИ должен поменять свое состояние. Но для перехода в другое состояние есть свои условия, как зависящие от внешних факторов, так и от самого ИИ.

В разработанной игре искусственный интеллект, как и игрок, управляет своим персонажем для забивания мяча в ворота противника. После анализа особенностей игрового процесса живых игроков на основе их поведения для

большей реалистичности были разработаны определенные состояния и переходы для ИИ (рис. 9).

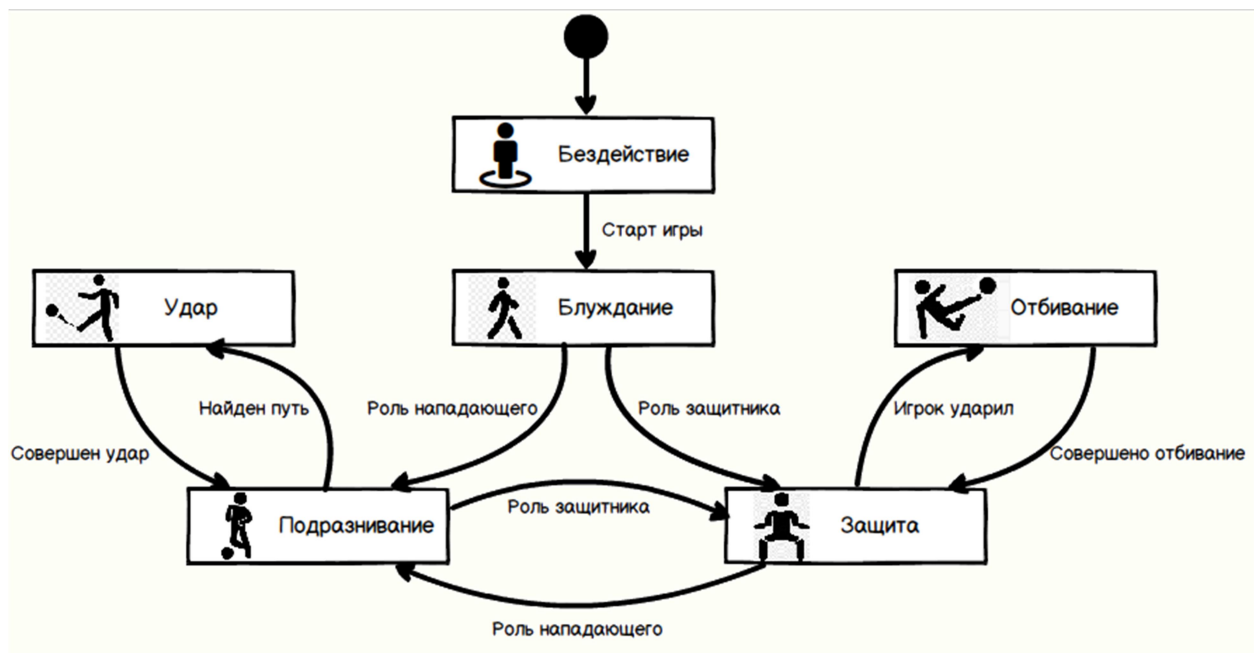


Рисунок 9. Схема состояний и переходов ИИ

- Idle (бездействие) – начальное состояние ИИ при запуске игры.
- SlowlyCircle (медленное блуждание вокруг стартовой позиции) – переход в это состояние осуществляется через секунду после начала игры.
- Tease («подразнивание» соперника, то есть создание иллюзии того, что вот-вот будет совершен удар) – переход в это состояние осуществляется при получении искусственным интеллектом роли нападающего, а также после каждого его удара по мячу.
- Hit (настоящий удар по мячу) – переход в это состояние осуществляется при доживании подходящего момента для удара, то есть когда соперник убрал своего персонажа с одного из лучших путей для мяча.
- BeatOff (попытка отбивания мяча под управлением соперника) – переход в это состояние осуществляется при ударе противника по мячу.
- Defend (быстрое и сконцентрированное блуждание по линиям перед мячом в ожидании удара соперника) – переход в это состояние осуществляется при получении искусственным интеллектом роли защитника, а также после каждого неудачного отбивания вражеского мяча.

Заключение

На основе анализа предметной области и изучения правил целевой настольной игры выявлены ее недостатки для цифрового аналога и обнаружены пути развития игровых условий, которые невозможны в реальном мире. Таким образом сформулированы задачи реализации проекта и выделены функции разрабатываемой игры.

Для решения поставленных задач в ходе разработки было проделано следующее:

- создан прототип с рабочим игровым процессом и интерфейсом;
- создана база данных и реализованы все связанные с ней функции;
- разработан мультиплеер для совместной игры;
- разработан искусственный интеллект для одиночной игры;
- осуществлено портирование на целевые платформы.

Результатом разработки игры стал продукт на платформах iOS и Android, отвечающий основным сформированным функциональным требованиям: игровой процесс согласно выработанным правилам, мультиплеер для совместной игры, искусственный интеллект для одиночной игры, связь с базой данных и портирование. Помимо игровой сцены было реализовано меню с необходимыми подразделами, имеющее эргономичный дизайн с футбольной тематикой.

Возможными перспективами развития проекта являются: соединение с необходимыми и распространенными онлайн-сервисами: для показа рекламных баннеров, на которых разработчик может заработать, для отслеживания более подробной статистики игроков и их достижений, вроде GameCenter. Также будут добавлены новые поля с большим пространством для перемещения игроков, новые эффекты, движения камер и многое другое. Планируется опубликовать игру в интернет-магазины Appstore и Google Play под крылом компании «Unnyhog FZ-LLC».

Список литературы

1. Страница продукта от разработчика Natenai Ariyatrakool «Glow Hockey» в интернет-магазине Google Play [Электронный ресурс]. URL: <https://play.google.com/store/apps/details?id=com.natenai.glowhockey> (дата обращения 23.03.2018).
2. Сайт игрового движка и средства для разработки игр Unity [Электронный ресурс]. URL: <https://unity3d.com/ru/> (дата обращения 14.02.2018).
3. Сайт хостинга beget [Электронный ресурс]. URL: <https://cp.beget.com> (дата обращения 07.03.2018).
4. Сайт онлайн-сервиса для импортирования моделей и анимаций Mixamo [Электронный ресурс]. URL: <https://www.mixamo.com> (дата обращения 05.04.2018).
5. Дейт К. Дж. Введение в системы баз данных, 8-е издание : пер. с англ. М. : Издательский дом «Вильямс», 2005. 1328 с.: ил. – Парал. тит. англ.
6. Коннолли Томас, Бегг Каролин. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание: пер. с англ. М. : Издательский дом «Вильямс», 2003. 1440 с.: ил.

List of references

1. Page of product “Glow Hockey” from developer Natenai Ariyatrakool in Google Play internet store. Mode of access:

<https://play.google.com/store/apps/details?id=com.natenai.glowhockey>, accessed March, 23, 2018.

2. Official website of game engine and game development platform Unity. Mode of access: <https://unity3d.com/ru/>, accessed February, 14, 2018.

3. Official website of hosting beget. Mode of access: <https://cp.beget.com>, accessed March, 07, 2018.

4. Official website of online service for importing models and animations Mixamo. Mode of access: <http://solutions.1c.ru/catalog/mdm>, accessed April, 05, 2018.

5. Date C. J. Introduction to Database Systems, 8-th Edition. Publishing House “Williams”, 2005.

6. Connoly Thomas, Begg Caroline, *Databases. Designing, development and maintenance. Theory and practice*, 3-rd edition, Publishing House “Williams”, 2003.

УСОВЕРШЕНСТВОВАНИЕ ЛАБОРАТОРНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ИСКЛЮЧЕНИЯ ВЛИЯНИЯ ЧЕЛОВЕЧЕСКОГО ФАКТОРА

УДК 004.92+66.012, ВАК 05.13.06, ГРНТИ 20.15.13

Усовершенствование лабораторной информационной системы для исключения влияния человеческого фактора

Improvement Of The Laboratory Information System For Eliminating The Effect Of The Human Factor

А.В. Бахановский, П.В. Плехов

A. V. Bakhanovskiy, P.V. Plekhov

ОАО «Березниковский содовый завод», г. Березники;
Пермский национальный исследовательский политехнический университет, г. Березники

Berezniki Sodium Plant
Perm National Research Polytechnic University, Berezniki

В статье отражено создание информационной системы позволяющей автоматически произвести расчет произведенной соды на ОАО «Березниковский содовый завод» и исключение человеческого фактора из процесса сбора информации для него. Рассмотрена деятельность сменных лаборантов корпусов, дежурного лаборанта и инженера производственно-технического отдела ответственного за расчет выработки. Описаны интерфейсы и функциональные составляющие ИС.

A design of information system for automatic calculation of sodium carbonate production accounting in Berezniki Sodium Plant is described. An activity of assistants, duty assistant and production department engineers is given. Interfaces and functions of the information system are designed and proved.

Ключевые слова: производство соды, OPC-server, база данных.

Keywords: sodium carbonate production, OPC-server, database.

Введение

В настоящее время на ОАО «Березниковский содовый завод» сбор данных для расчета выработки осуществляется следующим образом: сменным лаборантам по программе ЛИМС каждый час приходит задание. Они должны снять показания с расходомеров и занести их в сменный журнал. Процесс снятия выглядит следующим образом, лаборант звонит в цех, и начальник смены передает ему показания расхода с прибора. Затем лаборант заполняет задание, вносит полученные данные в журнал.

Лаборант (ЛИиК) раз в два часа получает задание в ЛИМС на проведение анализа рассола для определения количества связанного аммиака в нем. Данный процесс выполняется в ручную, занимает два человеко-часа. После лабораторного исследования рассола результат записывается в задание и заносится в сменный журнал.

Инженер (ПТО) получает задание на расчет выработанной соды от главного инженера. Затем запускает программу ЛИМС, заходит в раздел «Сменный журнал». Далее выбирает отчетный период, за который необходимо провести расчет. Экспортирует этот фрагмент журнала в EXCEL, форматирует его для удобства расчёта. И только после этих манипуляций с данными приступает к расчету по методике, принятой на предприятии. Данный расчет необходим для планирования объема выпуска продукции, расхода сырья, энергоресурсов.

После изучения процесса сбора информации для расчета и самого расчета выработки выявлены следующие проблемы:

- Сменный лаборант не всегда заносит достоверную информацию о расходе сырья, также он может проигнорировать срок выполнения задания и внести данные, когда пожелает;
- Возможности приборов в цехах не используются в полном объеме, а служат лишь простым дисплеем для снятия показаний;
- Инженер (ПТО) не может выполнить расчет оперативно, так как данные для него находятся в «сыром виде» и требуют предварительной подготовки;
- Инженер (ПТО) должен иметь хорошие навыки работы с EXCEL;
- Отчет по выработке соды за большой период времени очень трудоемкий и не достоверный;
- Результаты расчета представляются только в числах, что позволяет сделать выводы только о количестве выпущенной продукции.

Одним из вариантов решения проблем является внедрение специализированной информационной системы (ИС). Она должна:

- Опрашивать приборы КИП в реальном времени, тем самым исключить человеческий фактор при заполнении сменного журнала;
- Уменьшить время на проведение расчета;
- Позволять производить расчет в оперативном режиме;
- Иметь простой удобный интерфейс для работы инженера;
- Предоставлять несколько видов отчетов, для более эффективного управления производством.

Цель создаваемой информационной системы:

Уменьшение времени расчета выработки соды, исключение человеческого фактора при сборе данных.

Для достижения цели ИС решает следующие задачи:

- Производит автоматический расчет количества выработанной соды за отчетный период, извлекая информацию для расчетов из баз данных ;
- Представляет детальный отчет о расходе и качестве сырья за отчетный период;
- Представляет в графическом виде информацию о количестве подаваемой жидкости в колонны и наличии связанного аммиака в ней.

Программа представляет собой приложение, взаимодействующее с двумя базами данных (Рис.1) [1]. Программа рассчитана на ограниченное число пользователей.

Платформой ПС является *IBM-PC* совместимый компьютер с установленной на нем ОС семейства *Windows*, которые используются на всех рабочих местах завода.

Моделью ЖЦ программы стала спиральная модель [2]. На этапе внедрения мы имеем работоспособную ИС, но в дальнейшем заказчик хочет расширение функционала.

Выбор СУБД пал на продукт компании *Microsoft* - *Microsoft SQL-server* . Так как он позволяет хранить большие объемы информации, есть встроенные механизмы создания резервных копий БД [3]. Также на стороне заказчика имеются программисты, которые хорошо разбираются в *SQL*-запросах.

На предприятии заказчика в цехах по производству соды не существует SCADA системы. Соответственно информация с приборов не снимается и не хранится. Решено создать свою SCADA систему с помощью условно бесплатного программного обеспечения. И организовать хранение полученной информации. Приборы КИП подключить к вычислительной сети предприятия. И резервировать за каждым прибором, собственный IP-adress.

Так как одна из требуемых для расчета баз данных *I-LDS-bsz* уже существует на предприятии и размещена на *SQL-server* , то принято решение поместить вторую базу данных *Data_collect* на тот же *SQL-server*.(Рис.1)

Для опроса приборов, выбран *OPC-server* компании *Endress-Hauser*. Его преимущества в том, что он условно бесплатный имеет триальный режим на 30 дней и не требует большой вычислительной мощности. Но у него есть, и существенный недостаток: с ним не поставляется *OPC-client* [4].

В качестве *OPC-client* выбрана специальная надстройка для *EXCEL-OPCEx3* компании *Resolvica*. Данная надстройка является так же условно бесплатной. Имеет простой интерфейс, позволяет опрашивать приборы и записывать результаты напрямую в базу данных.

OPC-server и *OPC-client* устанавливается на сервер приложений предприятия. (Рис.1)

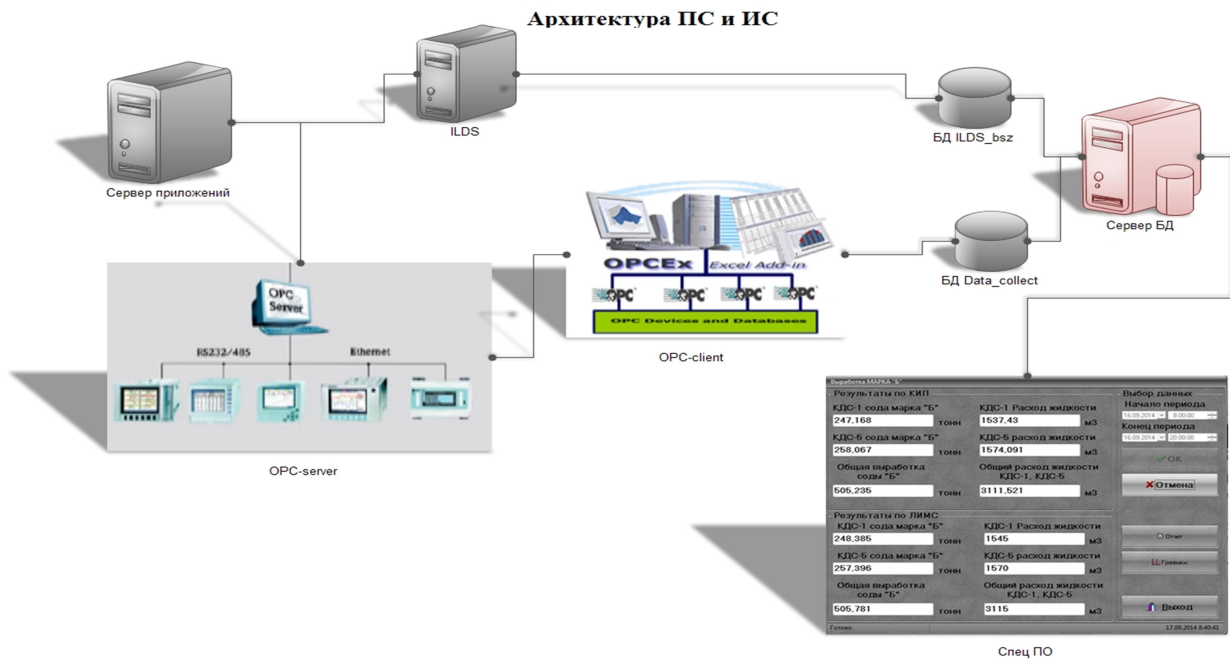


Рисунок 1. Архитектура ПС

При запуске программы открывается главный экран «Выработка марка «Б»» (Рис. 2). На этом экране активны только три элемента:

- установка даты начального периода;
- Кнопка «отмена»;
- Кнопка «выход».

Выработка МАРКА "Б"

Результаты по КИП		Выбор данных	
КДС-1 сода марка "Б"	КДС-1 Расход жидкости	Начало периода	01.09.2014 8:00:00
247.168 тонн	1537.43 м3	Конец периода	17.09.2014 20:00:00
КДС-5 сода марка "Б"	КДС-5 расход жидкости	<input type="button" value="OK"/> <input type="button" value="Отмена"/>	
268.067 тонн	1574.091 м3	<input type="button" value="Отчет"/> <input type="button" value="Графики"/>	
Общая выработка соды "Б"	Общий расход жидкости КДС-1, КДС-5	<input type="button" value="Выход"/>	
605.235 тонн	3111.521 м3		
Результаты по ЛИМС			
КДС-1 сода марка "Б"	КДС-1 Расход жидкости		
248.385 тонн	1545 м3		
КДС-5 сода марка "Б"	КДС-5 расход жидкости		
267.396 тонн	1570 м3		
Общая выработка соды "Б"	Общий расход жидкости КДС-1, КДС-5		
605.781 тонн	3115 м3		

17.09.2014 8:36:34

Рисунок 2. Главное окно программы

Это окно визуально разделено на три части:

- Выбор данных – в этой части пользователь может указать период расчета, и имеет возможность получить два вида отчетов. Также в этой части есть кнопка «выход» для окончания работы программы;

- Результат по КИП – здесь выводятся результаты расчетов по данным полученным из БД Data_collect, то есть без участия сменных (АБДСКБ1, АБДСКБ2) . По каждой колонне отдельно, так и общее количество произведённой соды и затраченного сырья ;

- Результаты по ЛИМС – здесь выводятся результаты расчетов по данным полученным из БД I-LDS-BSZ, которая заполняется лаборантами. Здесь также как и в части «Результат по КИП» выводятся результаты расчетов по каждой колонне отдельно, и общее количество произведённой соды и затраченного сырья. Эта часть еще необходима для того чтоб сравнить показания полученные ручным трудом и автоматическим. Тем самым позволяет контролировать работу сменных лаборантов.

Чтобы произвести автоматический расчет выработки соды необходимо выполнить следующие действия [5]:

- Выбрать начальный период расчета. Для этого необходимо в главном окне программы в разделе «Выбор данных» «Начало периода» с помощью выкидного календаря указать дату и время начального периода расчета . После этого активируется окно выбора конечного периода;
- Выбрать конечный период расчета. Производится аналогично выбору начального периода . После указания конечного периода расчета активируется кнопка «ОК». Это означает, что все параметры для начала сбора информации выбраны, можно приступить к опросу баз данных и выполнить расчет;
- Нажать кнопку «ОК» (Рис. 3).

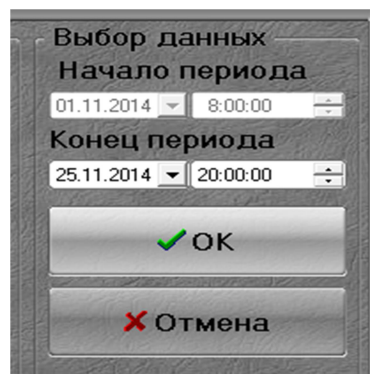


Рисунок 3. Готовность программы к расчету

После нажатия кнопки «ОК» программа начинает выполнение заложенных в нее функций.

Чтобы пользователь видел, на какой стадии находится расчет, предусмотрены два элемента:

- Progress bar – графически отображает ход вычислений (Рис. 4);

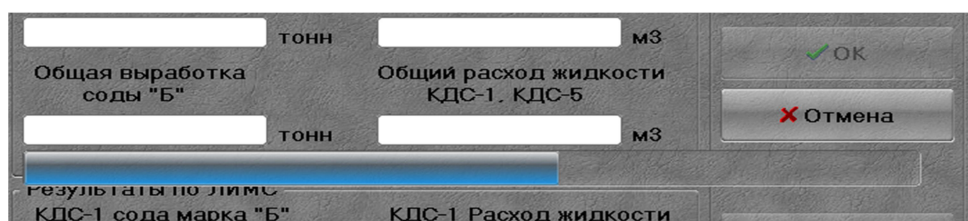


Рисунок 4. Графическое отображение хода вычислений

- Status bar – словесное описание выполняемых действий (Рис.5).

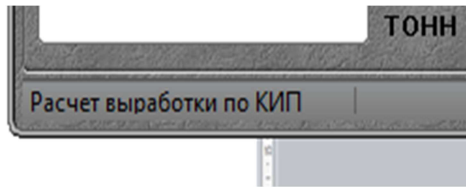


Рисунок 5. Словесное описание выполняемых действий

По мере выполнения расчетов заполняются соответствующие окна в главном окне программы «Результаты по КИП» и «Результаты по ЛИМС». Время выполнения расчета зависит от периода, указанного пользователем. В среднем расчет выработки соды за двенадцать часов занимает десять секунд.

Когда расчет будет произведен до конца, все окна в программе будут заполнены результатами. В строке «Status bar» появится надпись «Готово». Какой из результатов принимать к сведению для управления планом выпуска продукции, пользователь решает сам (Рис.6).

 Изображение представляет собой скриншот главного окна программы «Выработка МАРКА "Б"». Окно разделено на несколько секций. В центре расположены две таблицы с результатами расчетов. Справа находится панель «Выбор данных» с полями для выбора периода и кнопками «ОК», «Отмена», «Отчет», «Графики» и «Выход». В нижней строке окна отображается статус «Готово» и время «17.09.2014 8:40:41».

Результаты по КИП		Результаты по ЛИМС	
КДС-1 сода марка "Б"	КДС-1 Расход жидкости	КДС-1 сода марка "Б"	КДС-1 Расход жидкости
247.168 тонн	1537.43 м3	248.385 тонн	1545 м3
КДС-5 сода марка "Б"	КДС-5 расход жидкости	КДС-5 сода марка "Б"	КДС-5 расход жидкости
258.067 тонн	1574.091 м3	257.396 тонн	1570 м3
Общая выработка соды "Б"	Общий расход жидкости КДС-1, КДС-5	Общая выработка соды "Б"	Общий расход жидкости КДС-1, КДС-5
505.235 тонн	3111.521 м3	505.781 тонн	3115 м3

Рисунок 6. Вид главного окна после выполнения расчетов

Для того чтобы приступить к новому расчету необходимо нажать кнопку «Отмена». После ее нажатия вид главного окна вернется к первоначальному (Рис. 2).

По полученным расчетам выработки соды видно только численное выражение произведенной продукции и затраченного сырья. Этой информации недостаточно для того, чтобы проанализировать весь процесс производства. Для получения детального отчета о расходе и качестве сырья, израсходованного в

процессе выработки соды, в программе предусмотрена функция формирования отчета в виде книги EXCEL. Чтоб сформировать отчет необходимо выполнить следующие действия:

- Произвести расчет выработки соды. После того как расчет будет произведен, в главном окне программы станет активна кнопка «Отчет» (рис. 6);
- Нажать кнопку «Отчет».

Откроется книга EXCEL, содержащая шесть отчетов (Рис.7). По умолчанию книга откроется на листе с отчетом «КИП общая». Для выбора нужного пользователю отчета необходимо выбрать нужный лист из списка навигации по книге.

Жидкость на выходе из КДС-1/5						7	8	9
Время	КДС-1		Время	КДС-5				
	Расход м3/ч	Na.NH3 н.д		Расход м3/ч	Na.NH3 н.д			
16.09.2014 8:00	129.78	60.8	16.09.2014 8:00	127.49	62.4			
16.09.2014 9:00	120.08	60.8	16.09.2014 9:00	125.16	62.4			
16.09.2014 10:00	120.46	61.6	16.09.2014 10:00	128.67	61.2			
16.09.2014 11:00	129.79	61.6	16.09.2014 11:00	129.46	61.2			
16.09.2014 12:00	133.35	59.6	16.09.2014 12:00	130.37	63.2			
16.09.2014 13:00	133.89	59.6	16.09.2014 13:00	128.98	63.2			
16.09.2014 14:00	129.87	60.8	16.09.2014 14:00	129.39	62.8			
16.09.2014 15:00	129.73	60.8	16.09.2014 15:00	129.49	62.8			
16.09.2014 16:00	129.79	59.6	16.09.2014 16:00	130.18	60.4			
16.09.2014 17:00	129.88	59.6	16.09.2014 17:00	135.79	60.4			
16.09.2014 18:00	129.84	61.6	16.09.2014 18:00	139.58	61.2			
16.09.2014 19:00	120.97	61.6	16.09.2014 19:00	139.52	61.2			

Рисунок 7. Отчет

Отчеты делятся на два типа, по способу сбора информации:

- Отчеты ЛИМС – информация для этого типа отчета собирается из базы данных I-LDS-BSZ. Это информация о часовом расходе жидкости и наличии связанного аммиака в ней собрана лаборантами предприятия;
- Отчеты КИП – информация для этого типа отчетов собирается из двух баз данных. Информация о часовом расходе жидкости извлекается из базы данных Data_collect. Информацию о количестве связанного аммиака в жидкости получаем из базы данных I-LDS-BSZ.

Содержание отчетов:

- ЛИМС КДС-1 и ЛИМС КДС-5 – информация о часовом расходе жидкости и содержании связанного аммиака в ней по каждой колонне отдельно полученная из базы данных I-LDS-BSZ;
- ЛИМС общий – информация о часовом расходе жидкости и содержании связанного аммиака в ней общая по обеим колоннам;

- КИП КДС-1 и КИП КДС-5 - информация о часовом расходе жидкости и содержании связанного аммиака в ней по каждой колонне отдельно полученная из БД I-LDS-BSZ и Data_collect.

- КИП общий - информация о часовом расходе жидкости и содержании связанного аммиака в ней общая по обеим колоннам;

Для получения информации в виде графиков о количестве подаваемой жидкости на колонны и наличия связанного аммиака необходимо:

- Произвести расчет выработки соды. После того как расчет будет произведен на главной форме программы станет активна кнопка «Графики» (рис. 6);

- Нажать кнопку «Графики».

Откроется второе окно программы «Графики» (Рис. 8). В этом окне изображены шесть графиков.

Графики делятся на два типа по способу получения информации для построения:

- Графики, изображенные красным цветом – информация для их построения получена из БД Data_collect;
- Графики, изображенные зеленым цветом – информация для их построения получена из БД I-LDS-BSZ.

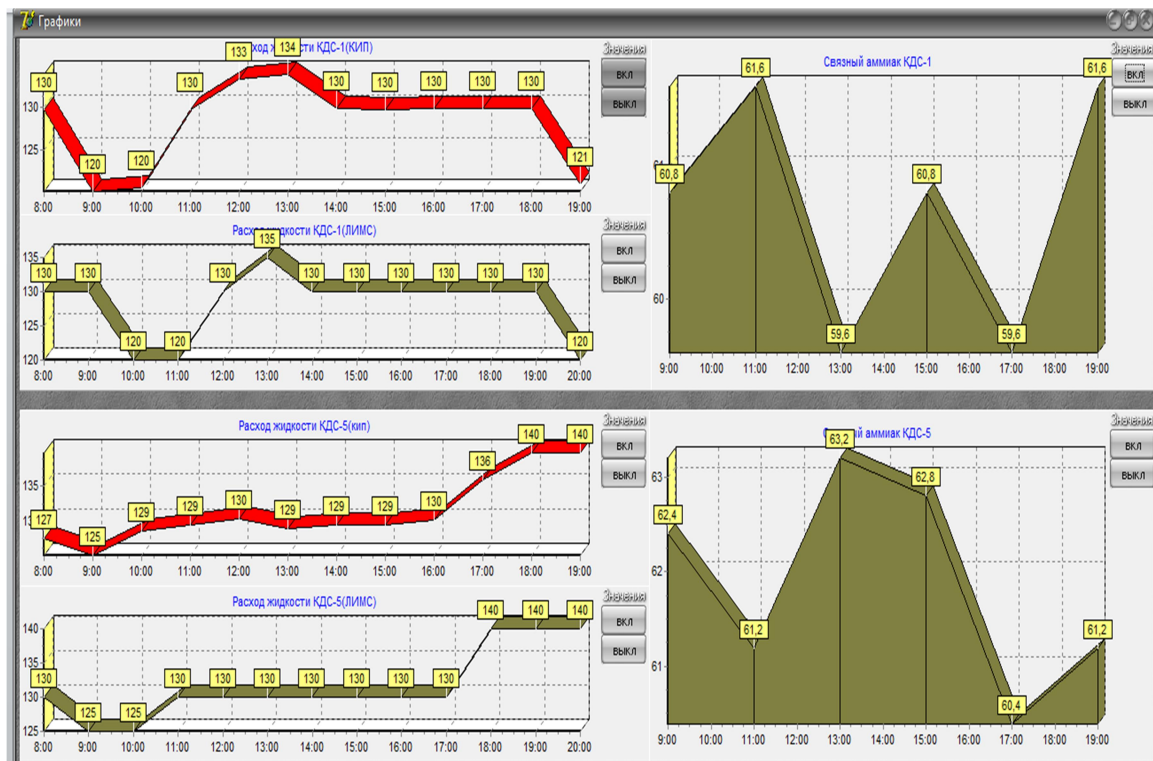


Рисунок 8. Окно «Графики»

На графиках изображено:

- Объем рассола, подаваемого в колонны (Рис.9);

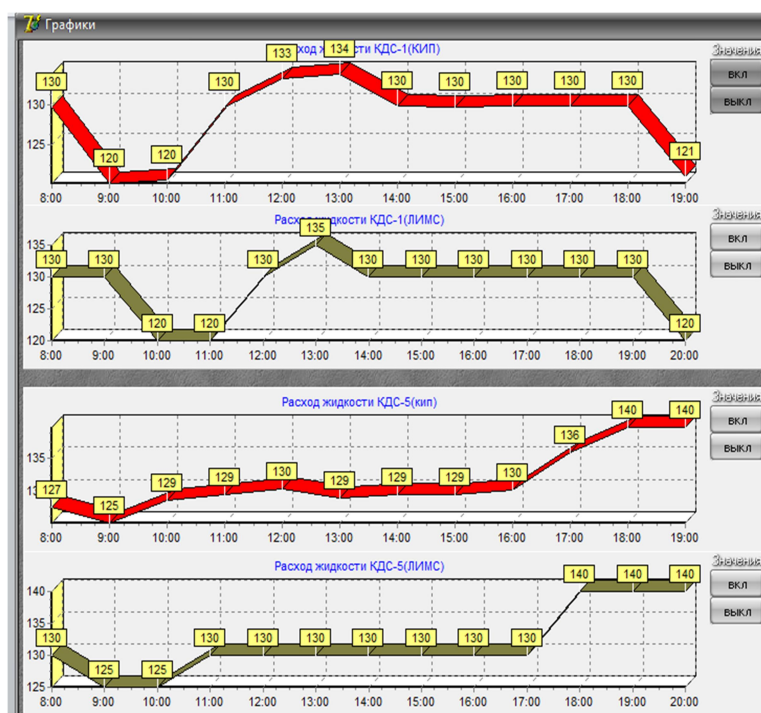


Рисунок 9. График расхода жидкости

- Количество связанного аммиака в рассоле (Рис. 10).

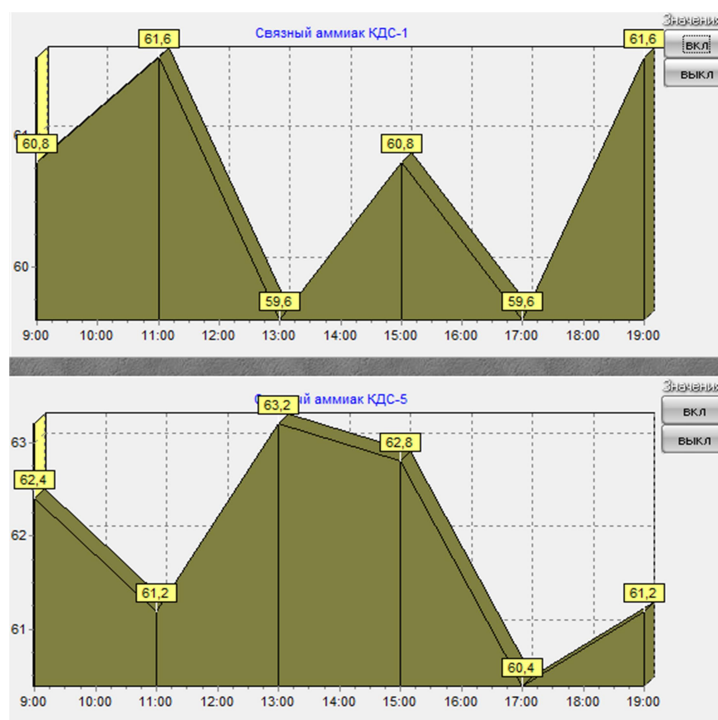


Рисунок 10. Количество связанного аммиака

Настоящая ИС может быть использована только на ОАО «Березниковский содовый завод». Но решения и приемы, использовавшиеся при ее создании, могут быть применены на любом предприятии. Количество одновременно работающих пользователей, ограничено числом работающего с данной программой персонала.

ИС используется для:

- рационального использования приборов КИП;
- автоматического сбора и хранения данных полученных с приборов;
- исключения человеческого фактора из процесса сбора информации;
- контроль над работой лаборантов предприятия;
- автоматического расчета выработки соды;
- получения актуальной информации о выполнении плана на предприятии;
- автоматического формирования отчета;
- уменьшения времени на выполнение расчета.

В процессе создания ИС все приборы КИП в цехах по производству соды были подключены к ЛВС предприятия и объединены в одну систему с помощью OPC-server. OPC-client: настроен на непрерывную запись данных в БД Data_collect. Что позволит контролировать работу цехов и прогнозировать выработку соды, а также позволит анализировать производительность колон.

Еще была проанализирована структура БД I-LDS-BSZ, и написаны SQL-запросы, для выборки из нее необходимой информации. Так как большое количество сотрудников предприятия заходило в программу ЛИМС только с целью получения данных о выработке соды, они занимали платные лицензии. Использование написанного приложения позволяет сократить количество платных лицензий для работы с программой ЛИМС на 10-15 единиц потому что, для получения этой информации больше не надо запускать ЛИМС.

Также принятые решения по применению условно бесплатного OPC софта, экономит средства на создание SCADA системы сторонней организацией.

Выводы

За счет использования такой ИС вся процедура расчета сводится к тому, что пользователь должен указать интересующий его период времени и нажать кнопку для запуска процесса расчета. Также время на выполнения задания инженером сократилось с одного часа до двух минуты. По окончании расчета пользователь получает автоматически сформированный отчет. Появилась возможность быстро рассчитать количество произведенной соды за любой отрезок времени. Сбор информации с приборов КИП происходит автоматически, что в свою очередь позволяет полностью исключить человеческий фактор из этого процесса. И позволяет снять эту обязанность со сменных лаборантов.

Список литературы

1. Бильфельд Н. В., Затонский А. В. Базы данных Реализация в системе Borland Delphi. Пермь: Перм. гос. тех. ун-т., 2006. 152 с.
2. Затонский А.В. Программные средства глобальной оптимизации систем автоматического регулирования. М.:ИНФРА-М, 2013. 136 с.
3. Кирин Ю.П., Затонский А.В., Беккер В.Ф., Бильфельд Н.В. Совместная разработка технологии и системы управления вакуумной сепарацией губчатого титана // Приборы и системы. Управление, контроль, диагностика. 2007. № 9. С. 7-10.

4. Кири́н Ю.П., Затонский А.В., Беккер В.Ф., Бильфельд Н.В. Качественный анализ динамики позиционного регулирования температуры процесса восстановления титана // Приборы и системы. Управление, контроль, диагностика. 2008. № 10. С. 54-56.
5. Затонский А.В. Компенсация скрытых стохастических свойств объекта автоматического регулирования // Известия Томского политехнического университета. 2010. Т. 316. № 5. С. 26-34.

List of references

1. Bilfeld N. V., Zatonsky A.V. Database Implementation in Borland Delphi system. Perm: Perm. state tech. Univ. of Illinois, 2006. 152 PP.2. Official website of game engine and game development platform Unity. Mode of access: <https://unity3d.com/ru/>, accessed February, 14, 2018.
2. Zatonsky A.V. Software for global optimization of automatic control systems. M.:INFRA-M, 2013. 136 PP.
3. Kirin Yu. P., Zatonsky A.V., Becker V. F., Bilfeld N. In. Joint development of technology and control system for vacuum separation of titanium sponge // Devices and systems. Management, control, diagnostics. 2007. No. 9. P. 7-10.
4. Kirin U. P., Zatonsky A.V., Bekker V. F., Bielfeld N. In. Qualitative analysis of the dynamics of positional adjustment of the temperature of the process of recovery of titanium // Devices and systems. Management, control, diagnostics. 2008. No. 10. Pp. 54-56.
5. Zatonsky A.V. Compensation of hidden stochastic properties of the object of automatic control. Izvestiya Tomsk Polytechnic University. 2010. Vol. 316. No. 5. P. 26-34.