

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Кафедра системного программирования

“Разработка эвристического алгоритма
высокоуровневого планирования проектов”

Дипломная работа студента 545 группы

Покалюка Юрия Алексеевича

Научный руководитель,
к.ф.-м.н.

_____/Д.Ю.Бугайченко/
/подпись/

Рецензент,
к.ф.-м.н., доц.

_____/А.Л.Тулупьев/
/подпись/

“Допустить к защите”
Заведующий кафедрой,
д.ф.-м.н., профессор

_____/А.Н.Терехов/
/подпись/

Санкт-Петербург
2009 год

Содержание

СОДЕРЖАНИЕ	2
1. ВВЕДЕНИЕ	3
1.1. ПРОБЛЕМАТИКА	4
1.1.1. Ошибочные прогнозные оценки затрат	4
1.1.2. Потеря накопленных знаний вследствие ухода сотрудников	5
1.2. ПРИМЕР	5
2. ПОСТАНОВКА ЗАДАЧИ	7
2.1. ЦЕЛИ И ЗАДАЧИ ДАННОЙ РАБОТЫ	7
2.2. ЦЕЛИ И ЗАДАЧИ ГРУППЫ РАБОТ	7
3. ОСНОВНЫЕ РЕЗУЛЬТАТЫ	8
3.1. ФОРМАЛЬНАЯ МОДЕЛЬ	8
3.1.1. Условие	8
3.1.2. Задача	8
3.1.3. Начальные и конечные условия	8
3.1.4. План	9
3.1.5. Следствия формальной модели	9
3.2. ЭВРИСТИЧЕСКИЙ АЛГОРИТМ ПЛАНИРОВАНИЯ ПРОЕКТОВ	10
3.2.1. Концепция алгоритма	10
3.2.2. Функции оценки полезности задач	10
3.2.3. Описание эвристического алгоритма планирования	11
3.2.4. Демонстрация работы алгоритма	12
3.2.5. Анализ эффективности	15
3.3. АРХИТЕКТУРА СИСТЕМЫ ПОДДЕРЖКИ ПЛАНИРОВАНИЯ ПРОЕКТОВ	18
3.3.1. Компоненты системы и взаимосвязи между ними	18
3.3.2. Структура классов системы	19
3.3.3. Возможности расширения системы	22
4. ЗАКЛЮЧЕНИЕ	23
5. СПИСОК ЛИТЕРАТУРЫ	24

1. Введение

Проект - это временное предприятие, предназначенное для создания уникальных продуктов, услуг или результатов при заданных ограничениях [21]. Таким образом, базовым свойством проектов является наличие целей и ограничений.

В настоящее время большинство компаний выстраивают деятельность вокруг реализации проектов. Например, строительные и юридические фирмы. Ключевым отличием проектно-ориентированных организаций является необходимость отдельно планировать каждый проект.

В рамках данной работы планирование проекта определяется через входящие в него процедуры:

- Формирование последовательности задач, приводящих к достижению целей (плана);
- Прогнозирование затрат на реализацию проекта (например, бюджета и продолжительности);
- Коррекция плана по ходу выполнения проекта.

Проектно-ориентированные организации можно ранжировать по степени уникальности реализуемых проектов:

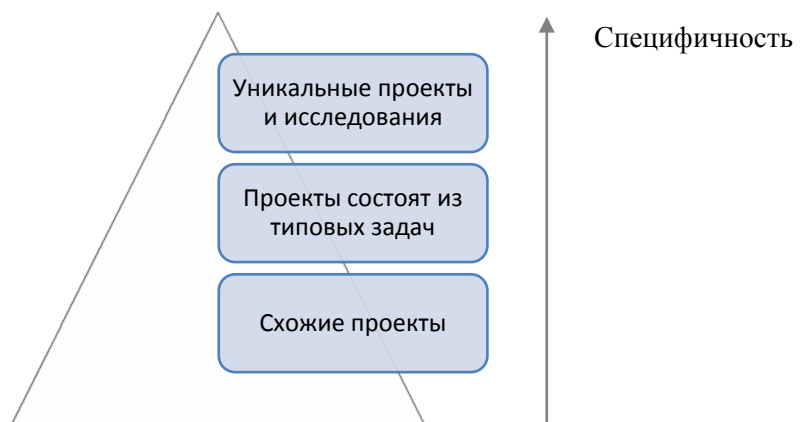


Рисунок 1. Категории проектов по степени уникальности

- **Схожие проекты** характеризуются повторяющимися от проекта к проекту последовательностями выполнения задач (например, строительство или издание книг). Таким образом, планирование в этом случае сводится к оценке затрат и последующей коррекции планов. Существует группа программных решений, ориентированных на схожие проекты: Instant Business Network [15], 1С Управление процессами [1] и Business Studio [3].

- *Проекты, состоящие из типовых задач*, можно разбить на типовые (для компании) блоки работ. Располагая пулом доступных задач, компания конструирует из них проекты.
- Основной особенностью *уникальных проектов* является непредсказуемость хода. Результаты каждого этапа могут существенно изменить концепцию проекта. Кроме того, в подобных проектах конкретные цели заменяются абстрактным и, зачастую, размытым видением результата. Классический пример – НИОКР.

1.1. Проблематика

На сегодня можно выделить 2 наиболее острые проблемы, связанные с планированием проектов:

- Слишком оптимистичные оценки затрат (что приводит к превышению бюджета и срыву расписания проекта);
- Зависимость от опытных сотрудников (носителей опыта планирования проектов).

Рассмотрим подробнее:

1.1.1. Ошибочные прогнозные оценки затрат

Согласно статистике, более 80% случаев превышения бюджета и/или сроков приходится на счёт ошибок планирования [25]. На практике, в большинстве случаев прогнозирование осуществляется одним из экспертов компании на основании его опыта. Основным минус экспертного прогнозирования – субъективность оценок. Аргументация принимаемых решений трудоёмка и, зачастую, эксперт полагается на интуицию и опыт.

Дополнением к экспертному прогнозированию является экстраполяция накопленных фактических данных (прогнозирование на базе статистики). Такой подход позволяет получить ориентировочные оценки затрат в случае, когда подобный проект уже выполнялся в компании.

Среди недостатков прогнозирования на базе статистики необходимо указать:

- Необходимость длительного накопления данных, прежде чем ценность их использования превысит затраты на сбор и аккумулирование;
- Затраты на накопление данных пропорциональны разнообразию проектов в компании.

1.1.2. Потеря накопленных знаний вследствие ухода сотрудников

Обычно знания делят на 2 группы: формализованные и персонализированные. Персонализированные знания покидают компанию вместе с их обладателем. Существует 2 способа предотвращения подобных потерь:

- Постоянное обучение сотрудников (передача опыта);
- Формализация знаний.

Формализация знаний и их накопление достаточно трудоёмка, вследствие чего нуждается как в эффективной организации, так и в инструментальной поддержке. При этом затраты на формализацию данных должны быть минимальными.

1.2. Пример

Рассмотрим консалтинговый проект по оптимизации бизнес-процессов компании, описанию показателей и разработке системы мотивации. На его примере отметим основные проблемы.

- Цели проекта:
 - сформировать модель “ТО-ВЕ”¹ бизнес-процессов компании;
 - разработать показатели эффективности деятельности;
 - разработать систему мотивации.
- Ограничения: задан бюджет и ограничения по срокам.

Результатом процесса планирования будет последовательность задач, приводящих к достижению целей:

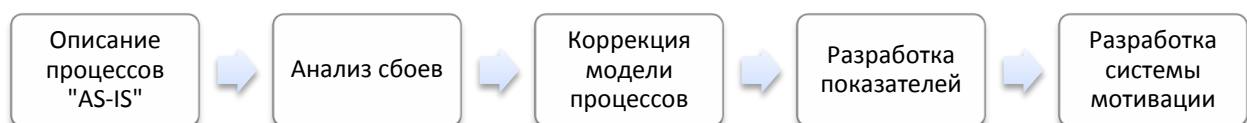


Рисунок 2. Пример. План проекта по оптимизации бизнес-процессов

Заметим, что, на практике, прогнозные оценки зачастую рассчитываются под давлением ограничений,

¹ Модель “ТО-ВЕ” (как должно быть) бизнес-процессов – схема, отражающая оптимальную для данной компании (и её целей), структуру бизнес-процессов.

Пути повышения эффективности планирования:

- Разработка альтернативных планов позволит выбирать между различными вариантами, а не “сжимать” привычный план до размера ограничений;
- Плановые оценки затрат должны определяться с учётом формализованного опыта реализации подобных задач.

2. Постановка задачи

2.1. Цели и задачи данной работы

Данная работа направлена на решение следующих задач:

1. Разработать эвристический алгоритм планирования проектов:
 - 1.1. Описать алгоритм;
 - 1.2. Проанализировать эффективность алгоритма.
2. Спроектировать систему поддержки планирования проектов:
 - 2.1. Описать связи между компонентами системы;
 - 2.2. Описать структуру классов;
 - 2.3. Описать возможные направления развития системы.

Данная работа (наравне с дипломными работами студентов 545 группы кафедры Системного программирования Математико-Механического факультета Санкт-Петербургского Государственного Университета Сергея Михайловича Курицына [5] и Уахата Алмасовича Бастимиева [2]) является частью общего проекта по разработке системы поддержки планирования проектов.

2.2. Цели и задачи группы работ

Группа работ направлена на разработку системы поддержки планирования проектов. В рамках достижения этой цели создаётся формальная модель системы, проводится анализ существующих решений в области автоматического планирования. На базе формальной модели и обзора решений выполняется проектирование и реализация системы.

3. Основные результаты

В данном разделе приводится описание модели, проделанной работы и достигнутых результатов.

3.1. Формальная модель

Система представляет собой пару $\langle C, T \rangle$:

- | | |
|----------------|--|
| Conditions (C) | - множество известных системе условий; |
| Tasks (T) | - множество известных системе задач; |

3.1.1. Условие ($c \in C$)

В каждый момент времени условие может быть либо выполнено, либо нет.

Состояние системы определяется как набор выполненных условий, т.е. $state = \{c_i\}$, $c_i \in C$.

3.1.2. Задача ($t \in T$)

Задача является основным объектом, с которым работает система. Каждая задача характеризуется двумя множествами условий: пред- и пост- условия ($t = \langle pre(t), post(t) \rangle$):

- $pre(t) \subset C$, набор условий, необходимых для выполнения данной задачи;
- $post(t) \subset C$, набор условий, выполняемых в рамках задачи.

Замечание: выполнение задачи расширяет текущее условие: задача выполняет новые условия, но не исключает уже выполненные.

Для перехода между состояниями (при выполнении задач) используется функция перехода: $F: 2^C \times T \rightarrow 2^C$:

$$F(\{c_i\}, t) = \{c_i\} \cup post(t)$$

3.1.3. Начальные и конечные условия

Как говорилось ранее, цель работы системы – предложение набора планов в соответствии с целями и ограничениями проекта. В базовой модели цели и ограничения задаются состояниями:

- Initial state (I) – начальное состояние, набор условий, выполненных к началу проекта;
- Goal state (G) – целевое (конечное) состояние, набор условий, которые должны быть выполнены по окончании проекта.

При этом конечное состояние не определено однозначно, оно подразумевает множество состояний GS:

$$GS = \{State_i\}: \forall i State_i \supset G$$

3.1.4. План

План – это последовательность задач, приводящая к выполнению всех условий, составляющих конечное состояние G. Каждая задача может быть выполнена по окончании предшествующей задачи:

$$plan = \{t_i\}, t_i \in T, \forall t_i pre(t_i) \subset I \cup \bigcup_{k=1}^{i-1} post(t_k)$$

$$G = \{c_i\} \subset (I \cup \bigcup_{k=1}^i post(t_k))$$

3.1.5. Следствия формальной модели

Процесс высокоуровневого планирования проектов и коррекции планов можно описать формальной моделью. Из приведённого описания можно выделить набор следствий, которые могут повлиять на эффективность реализуемых алгоритмов работы:

- В каждый момент времени система находится только в одном состоянии;
- По мере реализации плана состояние всегда только растёт (добавляются новые условия);
- Все переходы между состояниями последовательны.

3.2. Эвристический алгоритм планирования проектов

В рамках группы работ по разработке системы поддержки планирования проектов был реализован спектр различных алгоритмов конструирования проектных планов:

- Алгоритмы обхода графов (прогрессивные/регрессивные; в глубину/ширину);
- Адаптация алгоритма Дейкстры поиска кратчайших путей в ориентированном взвешенном графе;
- Эвристический алгоритм планирования.

В рамках данной работы был спроектирован и реализован эвристический алгоритм планирования.

3.2.1. Концепция алгоритма

Реализуемые алгоритмы направлены на поиск всех возможных планов. Результатом такого подхода является гарантированно лучший (при заданных ограничениях) план.

В рамках разработки эвристического алгоритма (далее – Алгоритма) рассматривается задача поиска набора планов, близких к оптимальному.

Эвристики – это правила выбора используемые для повышения эффективности обхода. Так как структура базы близка к графу, целесообразно опираться на модели обхода графа, но расширить его логикой выбора пути на каждом шаге. Эффективность эвристического алгоритма определяется логикой выбора порядка обхода вершин на каждом шаге – для этого введём функции оценки задач, позволяющие оценить перспективность путей.

3.2.2. Функции оценки полезности задач

Реализация данного алгоритма требует введения функций, оценивающих ценность имеющихся задач:

- Ценность задач определяется в контексте целей проекта и зависит от покрываемого данной задачей набора целей;
- Ценность задач снижается при появлении альтернатив (если одну цель можно достичь несколькими вариантами, ценность каждого из них убывает).

3.2.2.1. Оценка достижимости условий

Достижимость каждого условия можно характеризовать количеством задач, продуцирующих это условие. Кроме того, в контексте задачи планирования. Когда условие уже достигнуто (входит в начальные условия), оценка достижимости теряет актуальность. Формула расчёта достижимости условия:

$$c \in C, \quad A(c) = \begin{cases} |T_c|, T_c = \{t_i: \forall i \ c \in post(t_i)\}, \text{ если } c \notin I \\ -1, \text{ если } c \in I \end{cases}$$

3.2.2.2. Оценка ценности задач

Обладая наборами пред- и пост- условий, задача может рассматриваться как промежуточная цель, то есть характеризоваться как достижимостью, так и ценностью в контексте целей. Ценность задачи рассчитывается по формуле:

$$t \in T, \quad V(t) = \sum_{c \in post(t)} 1/A(c)$$

3.2.3. Описание эвристического алгоритма планирования

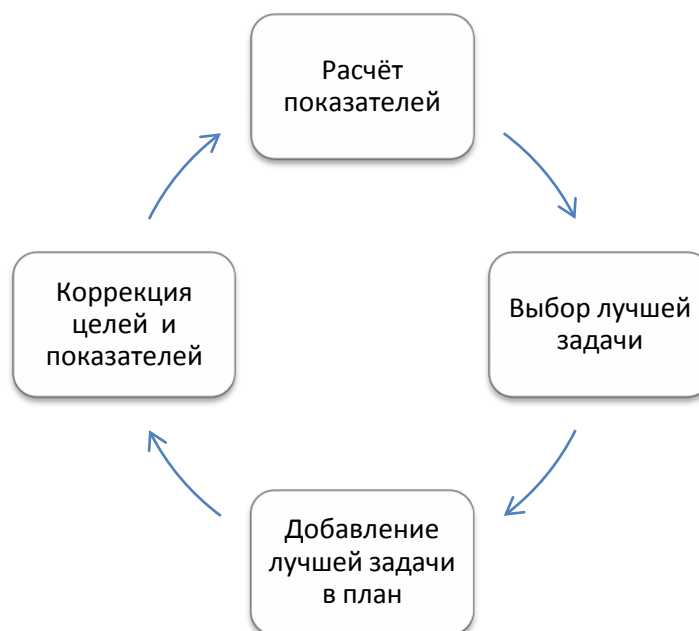


Рисунок 3. Схема работы эвристического алгоритма планирования

Алгоритм движется регрессивно в глубину, от целей к начальным условиям. На каждом шаге алгоритм выбирает лучшую (в контексте целей) задачу и добавляет её в план. После добавления происходит коррекция целей (достигнутые теряют актуальность, появляются новые – предусловия добавляемой задачи, за исключением предусловий, которые входят во множество начальных условий I). Коррекция списка целей влияет на оценки условий и задач:

- Выполнение задачи означает выполнение всех её пост- условий;
- Выполнение пост- условия уменьшает ценность задач, его продуцирующих, на

$$\Delta = 1/A(c)$$

- Аналогично, добавление новых целей влечёт увеличение ценности задачи на ту же величину.

Алгоритм работает итеративно до тех пор, пока не найдёт решения (список целей окажется пустым) или не докажет его отсутствия (если закончились возможные задачи).

3.2.4. Демонстрация работы алгоритма

Вернёмся к рассмотрению примера проекта по оптимизации бизнес-процессов, разработке показателей и системы мотивации (см. 1.2).

- Цели проекта:
 - C₁: сформировать модель “ТО-ВЕ”² бизнес-процессов компании;
 - C₂: разработать показатели эффективности деятельности;
 - C₃: разработать систему мотивации.
- Ограничения: задан бюджет и ограничения по срокам.

Далее по шагам приводится работа Алгоритма.

1. Находятся все такие задачи $t \in T$, что $\exists i: C_i \in \text{post}(t)$ (см. рис. 5: задачи T₁, T₂, T₃, T₄)

2. Каждое условие оценивается с точки зрения достижимости (см. 3.2.2.1);

Если находится условие, достижимость которого равна нулю (недостижимое), алгоритм возвращает сообщение об отсутствии решений. В рассматриваемом примере (см. рис. 5): $A(C_1) = 1$, $A(C_2) = 2$, $A(C_3) = 3$.

² Модель “ТО-ВЕ” (как должно быть) бизнес-процессов – схема, отражающая оптимальную для данной компании (и её целей), структуру бизнес-процессов.

3. Каждая задача оценивается с точки зрения ценности (см.3.2.2.2)

После оценки множество задач разбивается на 2: имеющие ценность ($V(t) > 0$) и остальные. $V(T_1) = 1,5$, $V(T_2) = 0,83$, $V(T_3) = 0,33$, $V(T_4) = 0,33$.

Ниже приводится схематичный рисунок, отражающий результат 3-х первых шагов:

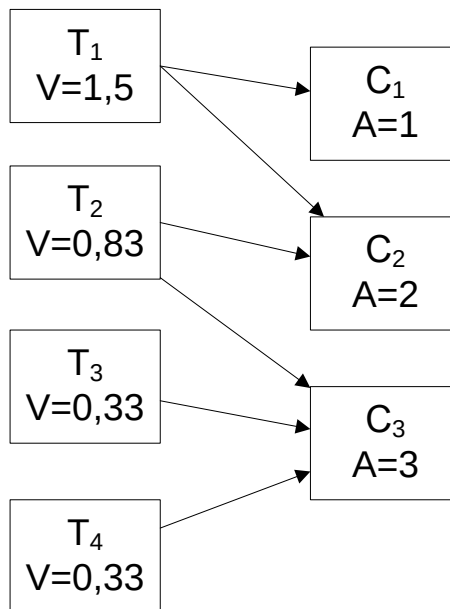


Рисунок 4. Эвристический алгоритм (шаги 1-3)

4. Выбирается наиболее ценная задача

В рассматриваемом примере – T_1 , она обладает наибольшей ценностью.

5. Корректируется набор целей

Выбор задачи T_1 приводит к тому, что условия C_1 и C_2 считаются выполненными, но к множеству G добавляются пред- условия задачи T_1 : C_4 и C_5

6. Корректируются оценки ценности задач

Изменение множества целей влечёт коррекции оценок задач:

- Ценность задачи T_2 снижается на 0,5 (так как $A(C_2) = 2$)
- Ценность задачи T_3 остаётся неизменной
- Из-за появления новых условий, в рассмотрение попадают задачи T_5 и T_6 ($V(T_5) = 1$, $V(T_6) = 0,5$)
- Ценность задачи T_4 увеличивается на 0,5 (из-за появления условия C_5)

Замечание: при добавлении задачи в план коррекции оценок достижимости условий не происходит.

После переоценки алгоритм возвращается в пункт 3.2.4.4.

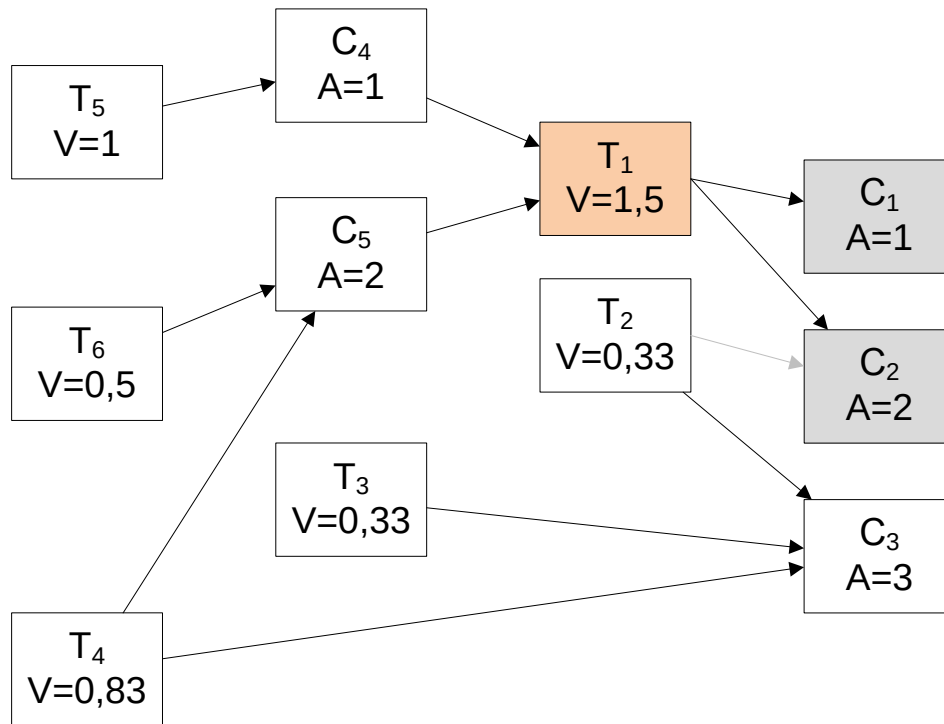


Рисунок 5. Эвристический алгоритм (шаги 4-6)

Возникающие в ходе работы алгоритма события:

7. Добавление недостижимой цели

Если добавление в план очередной задачи привело к появлению недостижимой цели (одно из пред- условий добавляемой в план задачи недостижимо, т.е. не существует задачи, содержащей его в числе своих пост- условий), значит, алгоритм пришёл в “тупик”. В этом случае из плана исключается последняя добавленная задача (при этом она исключается из рассмотрения и не может быть впоследствии добавлена в данный план). После исключения задачи, алгоритм корректирует оценки (процедура, обратная коррекции при добавлении задачи) и выбирает наиболее ценную задачу.

Замечание: при возвращении на шаг назад (удаление из плана задачи, приводящей в “тупик”) в число доступных возвращаются все задачи, которые были удалены на последующих шагах – удаление задачи из рассмотрения продолжается до того, как алгоритм вернётся на шаг, предшествующий её удалению.

3.2.5. Анализ эффективности

В данном разделе приводится сравнительный анализ Алгоритма с другими решениями, разработанными в рамках группы работ. Сравнение ведётся в контексте различных задач (начальные условия и конечные цели) на базе статистических оценок времени работы алгоритмов.

Эффективность работы алгоритмов варьируется в зависимости от структуры базы (задачи и связи между ними) и начальных данных. Таким образом, целесообразно проводить сравнение на различных классах задач. На основании описания Алгоритма можно судить о зависимости эффективности его работы от длины результирующего плана (количества задач, в него входящих): на каждом шаге Алгоритм корректирует оценки ценности задач.

Далее рассматривается динамика роста продолжительности работы алгоритмов при увеличении длины искомых путей и увеличению ветвистости задачи³. При этом учитывалось полное время работы алгоритма (выдача 3 близких к наилучшему планов).

3.2.5.1. Динамика продолжительности работы при увеличении длины искомых путей

Измерения велись по 3-м различным (по количеству задач) базам (значения приводятся в миллисекундах). В данной серии экспериментов использовались базы задач со слабоветвистой структурой, т.е. результирующие планы имеют 2-3 альтернативы.

База/ длина пути	Прямой обход в глубину	Прямой обход в ширину	Обратный обход в глубину	Алгоритм Дейкстры	Эвристический алгоритм
6/4	0,1638	0,04056	0,03744	0,0719	0,068452
10/5	0,22152	0,0468	0,05304	0,0525	0,10396
30/9	16,8113	35,8325	0,40404	0,5864	0,57252
30/14	6070,72	950,46	2,84	1,1486	0,8322

³ Под ветвлениями понимаются альтернативные пути, приводящие к достижению целей проекта

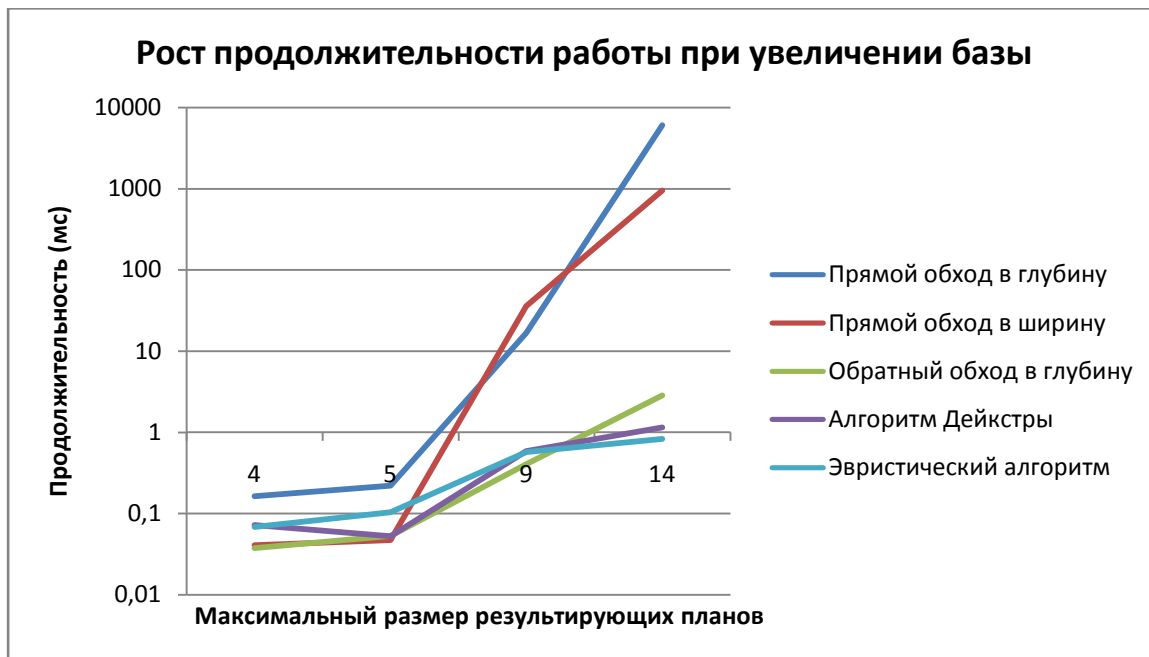


Рисунок 6. Рост продолжительности работы алгоритмов при увеличении базы

Как показывает диаграмма, эвристический алгоритм по производительности на слабоветвистых задачах близок к обратному обходу в ширину и алгоритму Дейкстры. Различия в продолжительности работы вызваны следующими причинами:

- На каждом шаге составления плана эвристический алгоритм корректирует оценки задач, вследствие чего его эффективность существенно зависит от количества задач в плане;
- Алгоритм регрессивного обхода графа не содержит подобных вычислений, из-за чего демонстрирует лучшую эффективность.

3.2.5.2. Динамика продолжительности работы при увеличении количества ветвлений

База/ кол-во ветвлений	Прямой обход в глубину	Прямой обход в ширину	Обратный обход в глубину	Алгоритм Дейкстры	Эвристический алгоритм
6/1	0,1638	0,04056	0,03744	0,0719	0,068452
10/5	0,25116	0,06864	0,06396	0,0936	0,10396
30/3	145,3125	124,2397	31,125	0,61258	0,54325
30/5	217,1638	174,5332	4,5324	2,58	0,82311

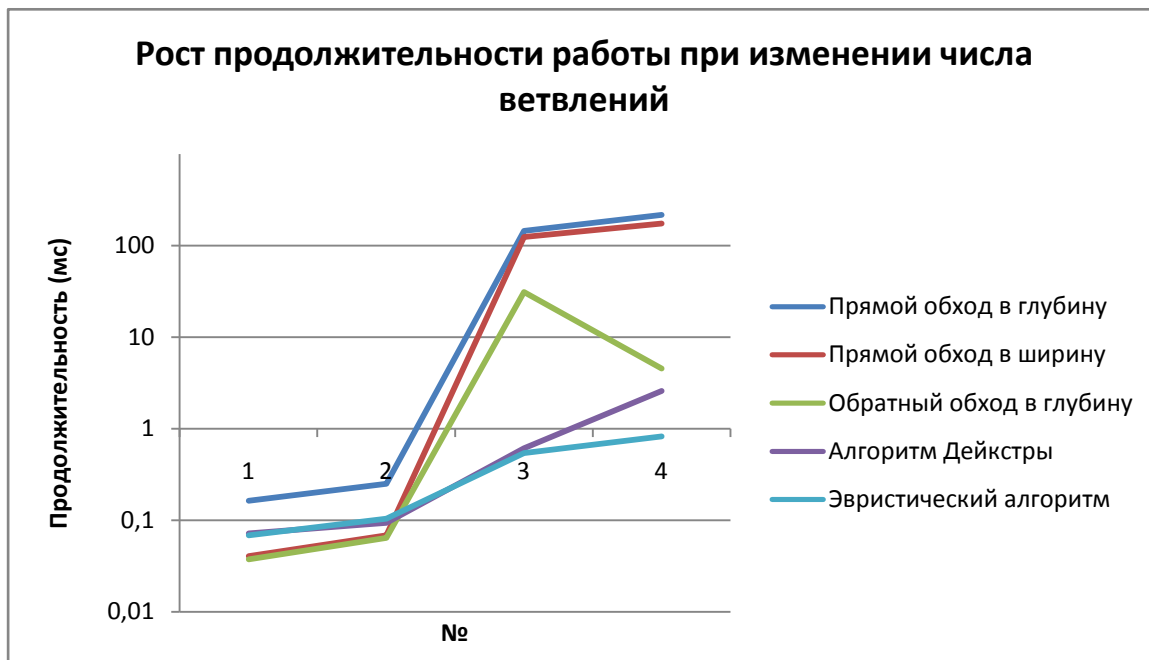


Рисунок 7. Рост продолжительности работы при изменении числа ветвлений

При увеличении количества ветвлений значительно возрастает продолжительность работы алгоритмов обхода графа. При этом эвристический алгоритм демонстрирует значительно лучшие результаты.

3.2.5.3. Выводы

Исходя из проведённых замеров, можно сделать выводы об эффективности эвристического алгоритма поиска проектных планов в сравнении с алгоритмами обхода графа. Разработанный алгоритм эффективен для:

- Задач с большим количеством альтернативных планов;
- Задач с множественными целями.

3.2.5.4. Пути повышения эффективности алгоритма

Основные временные потери связаны с пересчётом ценности задач и возвращением “на шаг назад” (удаление задачи из плана). В этих случаях повышения эффективности алгоритма можно добиться созданием дополнительных структур, в которых будут сохраняться текущие оценки.

3.3. Архитектура системы поддержки планирования проектов

Архитектура системы описывается от общего к частному, то есть сначала рассматриваются основные компоненты системы и взаимосвязи между ними, после чего представляется структура классов внутри компонент. Раздел завершается описанием возможных направлений расширения (развития) системы.

3.3.1. Компоненты системы и взаимосвязи между ними

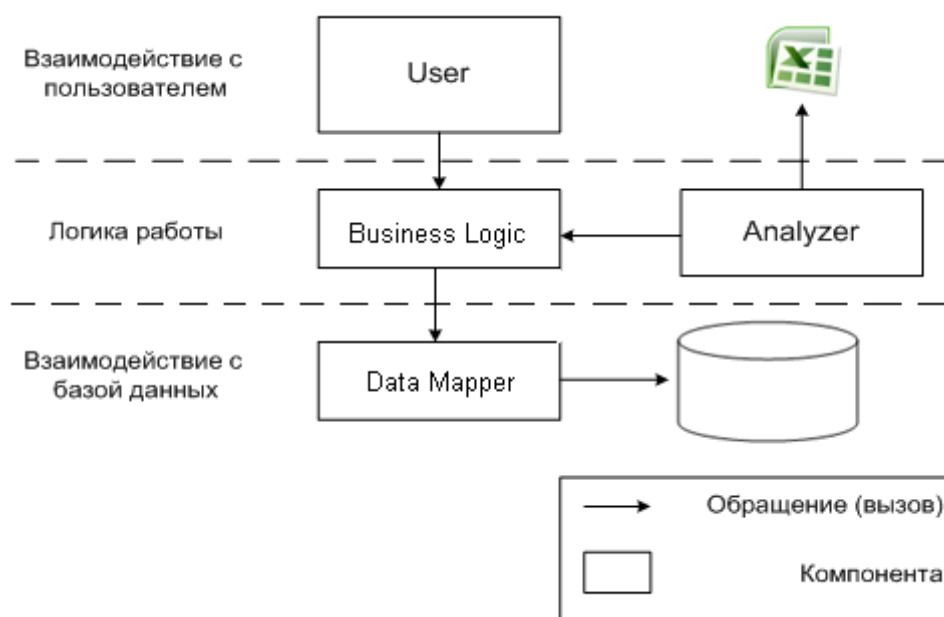


Рисунок 8. Компонентная структура решения

Система состоит из 3-х компонент:

- Business (общая логика работы, подробно описывается в разделе 3.3.2);
- User (графический интерфейс пользователя)
- Analyzer (приложение-анализатор эффективности алгоритмов)
- Data Mapper (доступ к данным)

Компоненты User и Analyzer подробно описаны в работе [2]. Компонента DataMapper описывается в работе [4].

3.3.2. Структура классов системы

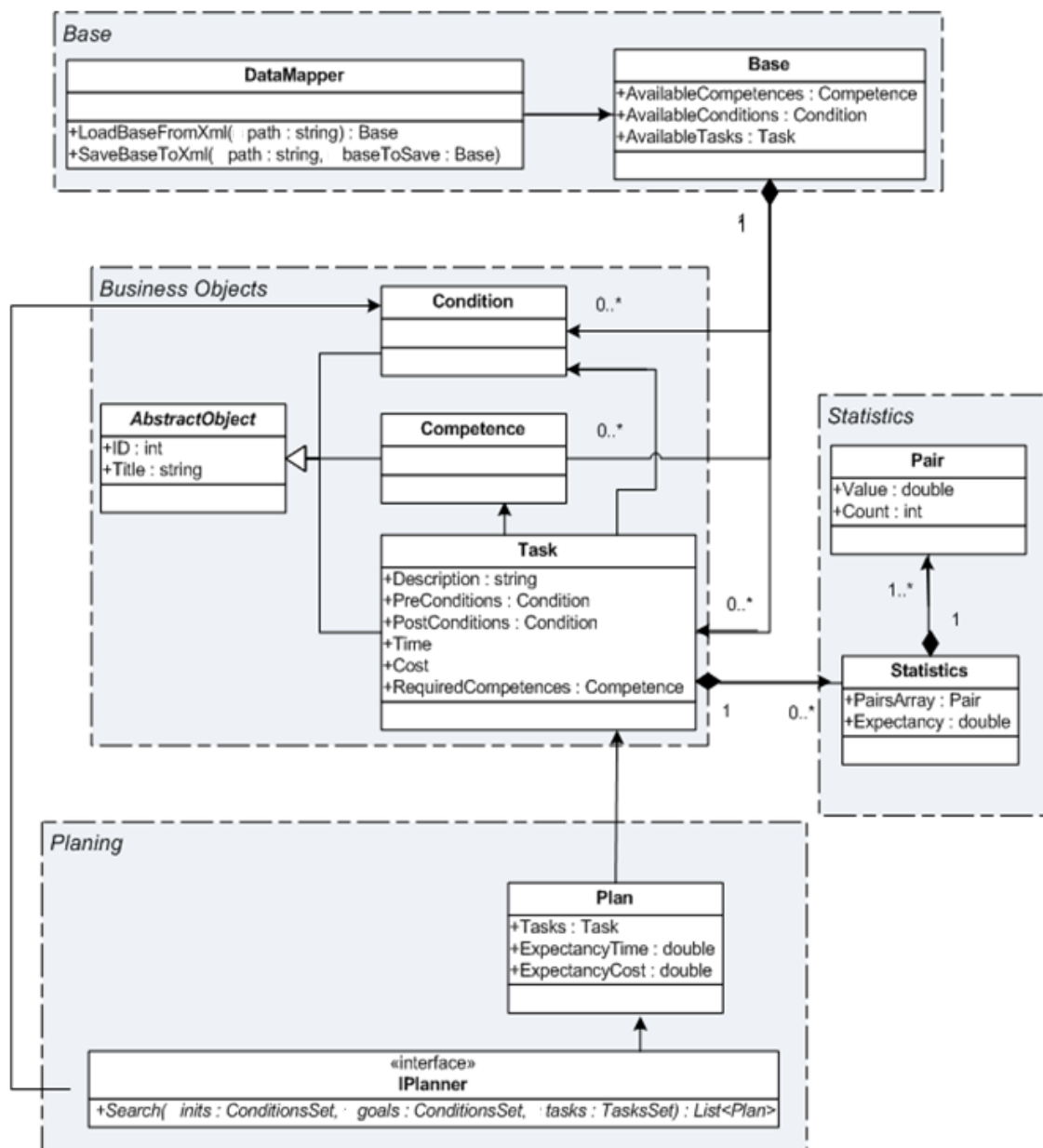


Рисунок 9. Диаграмма классов системы

Для удобства диаграмма классов рассматривается блоками:

3.3.2.1. Business Objects

Основные объекты, описанные в формальной модели. В системе выстроена иерархия наследования:

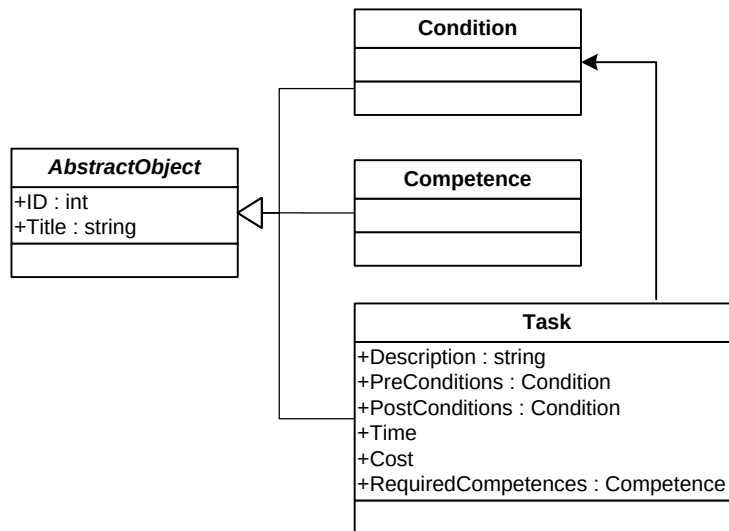


Рисунок 10. Фрагмент диаграммы классов – объекты формальной модели

Класс Condition описывает сущность Условие. Класс ConditionsSet (множество условий) реализует логику суммирования и вычитания условий (работа с состояниями). Методы соответствуют операциям объединения и вычитания множеств:

$$state_1 + state_2 = \{c_i\}: c_i \in state_1 \cup state_2$$

$$state_1 - state_2 = \{c_i\}: c_i \in state_1, c_i \notin state_2$$

Класс Task расширяет наследуемые параметры и содержит характеристики, описанные в формальной модели: списки пред- и пост- условий, компетенций, а также оценки продолжительности и стоимости.

3.3.2.2. Statistics

Вероятностная модель задаётся сопоставлением результату количества экспериментов, когда данный результат был достигнут (класс Pair).

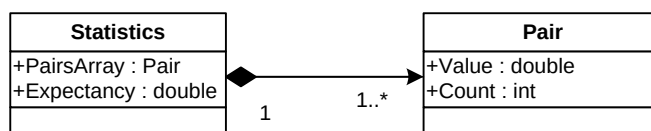


Рисунок 11. Фрагмент диаграммы классов – Статистика

Класс Statistics объединяет пары “результат - количество экспериментов” и рассчитывает математическое ожидание результата эксперимента. Подробно используемая в системе вероятностная модель описывается в [5].

3.3.2.3. Base

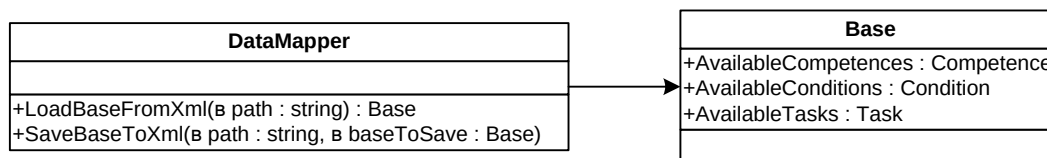


Рисунок 12. Фрагмент диаграммы классов - работа с базой

Для работы с базой данных использует экземпляр класса **Mapper**, который реализует методы загрузки и сохранения объектного представления в XML базу данных.

3.3.2.4. Planing

Вследствие того, что система использует несколько алгоритмов планирования, в соответствии с паттерном проектирования Strategy [4] был реализован общий интерфейс **IPlanner**.

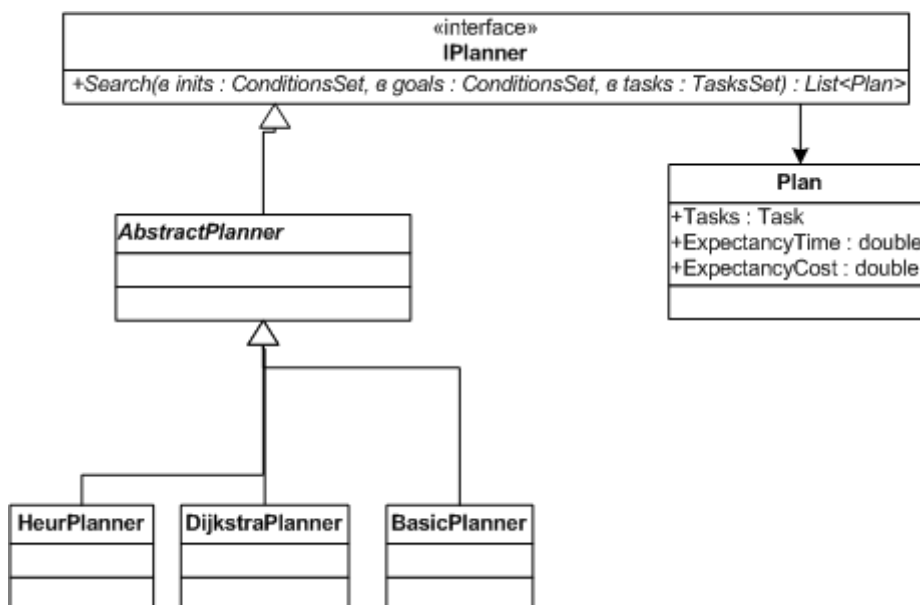


Рисунок 13. Фрагмент диаграммы классов - Поисковые классы

В ходе процесса проектирования был создан общий интерфейс **IPlanner**, определяющий сигнатуру общего для всех классов-планировщиков метода `Search`.

Данный метод получает списки пред- и пост- условий и набор доступных задач, а в результате работы возвращает список сконструированных планов.

В ходе реализации различных алгоритмов планирования были выявлены общие вспомогательные методы – они были реализованы в абстрактном классе `AbstractPlanner`, который наследуется классами-планировщиками.

3.3.3. Возможности расширения системы

Предложенная архитектура системы предполагает различные возможности развития реализованного решения:

- Расширение спектра характеристик задач и реализация дополнительной функциональности, которая позволит пользователю самому создавать требуемые характеристики и области их значений.
- Интеграция с другими инструментами поддержки проектного управления (например, Microsoft Project):
 - Анализ завершённых (и описанных) проектов и аккумуляция фактической информации в единую базу.
- Дополнительная визуализация связей между задачами.

4. Заключение

В рамках данной работы были выполнены все поставленные задачи:

1. Разработан эвристический алгоритм планирования проектов:
 - 1.1. Описан алгоритм;
 - 1.2. Проведён анализ эффективности алгоритма.
2. Описана архитектура системы поддержки планирования проектов:
 - 2.1. Описаны связи между компонентами;
 - 2.2. Описана структура классов;
 - 2.3. Описаны возможности расширения системы.

Таким образом, все поставленные в рамках работы задачи были успешно выполнены.

В рамках выполнения группы работ была достигнута основная цель - **разработан рабочий прототип системы поддержки планирования проектов**: прототип передан в опытную эксплуатацию (акт приёмки в опытную эксплуатацию прилагается к дипломной записке).

Кроме того, на базе группы работ по разработке системы поддержки планирования проектов был подготовлен доклад, который был представлен 27 мая 2009 года на конференции ИММВИИ-2009 (Интегрированные модели, мягкие вычисления, вероятностные системы и комплексы программ в Искусственном Интеллекте) в г. Коломна. Доклад был включён в сборник трудов конференции.

5. Список литературы

1. 1С Управление процессами // ПитерСофт. - <http://www.piter-soft.ru/36.html>.
2. **Бастимиев Уахат Алмасович** Адаптация поиска пути в графе для решения задачи высокоуровневого планирования проектов. - Санкт-Петербург, 2009.
3. Бизнес студио 2.0. - <http://www.businessstudio.ru>.
4. **Гамма Э., Хелм. Р, Джонсон Р., Влиссидес Дж.** Приемы объектно-ориентированного проектирования. Паттерны проектирования, издательство Питер, ISBN 978-5-469-01136-1
5. **Курицын Сергей Михайлович** Адаптация алгоритма Дейкстры к задаче высокоуровневого планирования проектов. - Санкт-Петербург., 2009 г..
6. **Линдгрэн Матс и Бандхольд Ханс** Сценарное планирование. - Москва : Олимп-Бизнес, 2009. - ISBN 978-5-9693-0137-5.
7. **Шеер Август-Вильгельм** Моделирование бизнес-процессов. - Москва : Весть-метаТехнология, 2000. - ISBN 5-89163-049-4.
8. **Avrim L. Blum Merrick L. Furst** Fast Planning Through Planning Graph Analysis // Artificial Intelligence. - 1997 г.. - 90.
9. **Bone B. Geffner H.** Planning as Heuristic Search: New Results. - Berlin : Springer-Verlag, 1999.
10. **Collis David J. и Montgomery Cynthia A.** Corporate strategy: a resource-based approach. - Boston : McGraw-Hill, Ernst & Young, 2005. - ISBN 0-07-231286-6.
11. **Fikes R. Nilsson N.** Strips: A new approach to the application of theorem proving to problem solving // Artificial Intelligence. - 1971 г.. - 2.
12. **Goodpasture John** Rolling Wave Planning // PM Boulevard. - 2008 г.. - <http://www.pmboulevard.com/Default.aspx?page=View%20Content&cid=1644&parent=Content%20Map>.
13. **Henry Kautz Bart Selman** Blackbox: A New Approach to the Application of Theorem Proving to Problem Solving. - 1998.
14. **Hoffmann J. Nebel B.** The FF Planning System: Fast Plan Generation through Heuristic Search // Journal of Artificial Intelligence Research. - 2001 г.. - 14.

15. IBN. - компания "Mediachase". - <http://pmbox.ru/business-process-management.aspx>.
16. Intro To Project Management for Entrepreneurs - <http://www.onixsoft.com/story3588.html>.
17. **Kragic Danica** Planning. - 2008 r..
18. **Kutluhan Erol James Hendler, Dana S. Nau** Complexity Results for HTN Planning // Annals of Mathematics and Artificial Intelligence. - 1996 r..
19. **Nau Dana S.** Current Trends in Automated Planning // AI Magazine. - 2007 r.. - 28.
20. **Penberthy J. S., Weld, D.** UCPOP: A Sound, Complete, Partial-Order Planner for ADL. [Конференция]. - San Francisco : Morgan Kaufmann Publishers, 1992.
21. **PMI, Inc.** PMBoK, Project Management Body of Knowledge. - Newtown Square, Pennsylvania 19073-3299 USA : Project Management Institute, Inc., 2004. - ISBN 1-930699-77-8.
22. **Rothman Johanna** Starting With Rolling Wave Planning // Rothman Consulting Group, Inc.. - 2007. - <http://www.jrothman.com/Papers/rolling-wave-planning.html>.
23. **Shewhart Walter Andrew** Statistical Method from the Viewpoint of Quality Control [Книга]. - New York : Dover, 1939. - ISBN 0-486-65232-7.
24. **T. Estlin D. Gaines, C. Chouinard, R. Castano, B. Bornstein, M. Judd, I. Nesnas, R. Anderson** Increased Mars Rover Autonomy using AI Planning, Scheduling and Execution - Рим : IEEE International Conference on Robotics and Automation (ICRA 2007), 2007.
25. **Wickham Philip** Management Consulting : Pearson Education Limited, 2004. - ISBN 0-273-68347-0.