

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
Саратовский государственный технический
университет имени Гагарина Ю. А.**

***Институт прикладных информационных
технологий и коммуникаций***

**Отчет по практической работе №3
«Обоснование причинно-следственных связей между переменными»**

по курсу «Организация, управление, планирование
и прогнозирование научных исследований»

Выполнили студенты группы МИФСТ-11:

Селютин А.Д,
Большелапов М.А,
Зайцев Е.П.

Проверил: Кушников В.А.

Саратов 2021

Моделируемые переменные и возмущения

Исследуемые показатели

На основе выбранных ранее 15 параметрах из ГОСТ Р ИСО/МЭК 9126-2001, таких как:

- $L_1(t)$ - надежность (reliability);
- $L_2(t)$ - практичность (usability);
- $L_3(t)$ - эффективность (efficiency);
- $L_4(t)$ - сопровождаемость (faintainability);
- $L_5(t)$ - защищенность (security);
- $L_6(t)$ - согласованность системы в целом (cofpliance);
- $L_7(t)$ - завершенность (faturity);
- $L_8(t)$ - анализируемость (analysability);
- $L_9(t)$ - изменяемость (changeability);
- $L_{10}(t)$ - стабильность (stability);
- $L_{11}(t)$ - тестируемость (testability);
- $L_{12}(t)$ - простота установки (installability);
- $L_{13}(t)$ - устойчивость к ошибкам (faulttolerance);
- $L_{14}(t)$ - восстанавливаемость (recoverability);
- $L_{15}(t)$ - понятность (understandability).

Будет производиться построение системы зависимостей.

Возмущения

Результаты комплекса мероприятий, необходимых для поддержания требуемого уровня качества у программного обеспечения интеллектуальных систем, показывают, что в качестве возмущений (внешних факторов) в модели целесообразно использовать следующие показатели:

- $q_1(t)$ - опыт разработчиков программного комплекса;
- $q_2(t)$ - опыт работы эксплуатационного персонала;

- $q_3(t)$ - трудоемкость разработки программного обеспечения;
- $q_4(t)$ - курс рубля по отношению к доллару и евро, соответственно;
- $q_5(t)$ - деловая репутация организации, в которой осуществляется эксплуатация программного комплекса.

Граф причинно-следственных связей

Граф причинно-следственных связей между моделируемыми переменными $L_i(t), i = \overline{1,15}$ и факторами внешней среды $q_i(t), i = \overline{1,5}$ формируется с разбиением на отдельные подграфы $G_{m_i}, i = \overline{1,15}$, каждый из которых используется при формировании соответствующего нелинейного дифференциального уравнения.

Матрица инцидентностей графа представляет собой матрицу $A(|L + q|)$ размером 15×20 по числу моделируемых переменных $L_i(t), i = \overline{1,15}$ и возмущений $q_i(t), i = \overline{1,5}$.

Значения элементов этой матрицы определяются следующими выражениями:

1. $\forall i \leq 20, \forall j \leq 20 a_{ij} = +1$, если увеличение значения переменной $L_i(t), i = \overline{1,15}$ или фактора внешней среды $q_i(t), i = \overline{1,5}$ приводит к увеличению переменной $L_i(t), i = \overline{1,15}$ или фактора внешней среды $q_i(t), i = \overline{1,5}$.
2. $\forall i \leq 20, \forall j \leq 20 a_{ij} = -1$, если увеличение значения переменной $L_i(t), i = \overline{1,15}$ или фактора внешней среды $q_i(t), i = \overline{1,5}$ приводит к уменьшению переменной $L_i(t), i = \overline{1,15}$ или фактора внешней среды $q_i(t), i = \overline{1,5}$.
3. При отсутствии связи между указанными переменными и факторами $a_{ij} = 0$.

Связи графа (подграфы) причинно-следственных связей, определяющие взаимосвязи между переменными формируемой системы, приведены в

таблице 1. По горизонтали указаны исследуемые переменные, по вертикали зависимости ранее описанных переменных от других.

Значения элементов таблицы выбраны в соответствии с мнением экспертов о релевантности причинно-следственных связей, влияющих на моделируемую переменную. Эти значения могут быть изменены при внедрении разрабатываемого математического обеспечения для требуемого уровня качества программного обеспечения интеллектуальных систем на конкретном предприятии.

Таблица 1 – Матрица графа причинно-следственных связей, определяющий зависимости, влияющие на исследуемые показатели $L_i(t)$

	L₁	L₂	L₃	L₄	L₅	L₆	L₇	L₈	L₉	L₁₀	L₁₁	L₁₂	L₁₃	L₁₄	L₁₅
L₁	0	0	0	1	1	1	0	0	0	1	0	0	1	1	0
L₂	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
L₃	0	1	0	0	0	1	0	0	1	0	0	1	1	0	1
L₄	0	0	0	0	0	1	1	1	-1	0	1	1	1	0	1
L₅	1	0	1	1	0	0	0	-1	-1	0	-1	0	1	0	0
L₆	1	0	1	1	1	0	1	1	1	1	1	0	1	0	1
L₇	1	0	1	1	1	1	0	1	-1	0	-1	0	0	1	0
L₈	0	0	0	1	1	1	0	0	-1	0	1	0	0	0	1
L₉	0	0	0	-1	-1	-1	-1	-1	0	-1	0	-1	0	0	1
L₁₀	1	0	1	1	1	0	0	0	-1	0	1	0	1	1	0
L₁₁	0	0	0	1	1	0	0	1	0	1	0	0	0	0	1
L₁₂	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
L₁₃	1	1	1	0	1	0	1	1	0	1	1	0	0	1	0
L₁₄	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0
L₁₅	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0
q₁	1	0	0	0	1	1	0	0	1	0	1	1	1	0	1
q₂	1	1	1	1	1	0	0	0	0	1	0	1	1	1	1
q₃	-1	0	0	0	1	1	1	1	1	0	1	1	0	1	-1
q₄	0	-1	0	0	0	0	1	0	1	0	0	0	-1	0	0
q₅	0	1	-1	0	0	0	0	0	0	1	0	1	0	-1	1

Обоснование причинно-следственных связей между переменными

Прежде чем перейти к обоснованию наличия или отсутствия причинно-следственных связей между переменными модели необходимо отметить, что ниже будут рассматриваться только релевантные с точки зрения разработчика ПО связи, т.к. в силу закона всеобщей связи явлений на качество программного обеспечения влияет огромное количество событий реального мира. Таким образом, учитываемые ниже связи представляют собой лишь сравнительно небольшое по мощности подмножество множества всех связей, выбранное в соответствии с предпочтениями исследователя.

Для переменных между которыми выполняется отношение «зависит-от» введем обозначение: $L_i(t)RL_j(t)$.

$L_1(t)$ - Надежность

Под надежностью $L_1(t)$ понимают способность программного продукта безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.

На надежность программного обеспечения ($L_1(t)$) влияют такие переменные как:

- Защищенность программного обеспечения (переменная $L_5(t)$), так как надежность ПО невозможно обеспечить без достаточного уровня защищенности системы при возможных информационных атаках на последнюю. Справедливо отношение $L_1(t)RL_5(t)$;
- Согласованность системы в целом ($L_6(t)$), так как при некорректной работе взаимосвязанных модулей невозможно будет обеспечивать выполнение функций с достаточной точностью. Справедливо отношение $L_1(t)RL_6(t)$;
- Завершенность ПО ($L_7(t)$) позволяет обеспечивать надежность системы, так как весь функционал разрабатываемого ПО будет реализован и проверен;

- Стабильность программного обеспечения ($L_{10}(t)$) напрямую связана с надежностью, так как функции должны выполняться точно при разных нагрузочных данных. Выполняется отношение $L_1(t)RL_{10}(t)$;
- Устойчивость системы к ошибкам ($L_{13}(t)$), как и стабильность, непосредственно связаны с надежностью системы. Для данной зависимости справедливо $L_1(t)RL_{13}(t)$;
- Восстанавливаемость системы ($L_{14}(t)$), связана с надежностью системы, поскольку возможность системы работать корректно после сбоя влияет на выполнение функций системы. Для данной зависимости справедливо отношение $L_1(t)RL_{14}(t)$.

$L_2(t)$ - Практичность

Практичность ПО $L_2(t)$ формируется из таких свойств как понятность, простота использования, возможность изучения и привлекательность программного средства.

На практичность ($L_2(t)$) влияют такие переменные как:

- Эффективность программного обеспечения (переменная $L_3(t)$), так как пользователи системы должны извлекать пользу, эффективно взаимодействуя с системой. Справедливо отношение $L_2(t)RL_3(t)$;
- Простота установки ПО ($L_{12}(t)$), поскольку развертывание системы является подпроцессом непосредственной эксплуатации;
- Устойчивость к ошибкам ($L_{13}(t)$) позволяет обеспечивать комфортное взаимодействие с системой и влияет на практичность использования разрабатываемого ПО. Справедливо отношение $L_2(t)RL_{13}(t)$;
- Восстанавливаемость программного обеспечения ($L_{14}(t)$) оказывает влияние на практичность системы, так как после возможных сбоев и отказов работы системы, необходимо восстанавливать данные последней работы ПО. Выполняется отношение $L_2(t)RL_{14}(t)$;

- Понятность системы ($L_{15}(t)$), непосредственно связана с ее практическим применением. Для данной зависимости справедливо отношение $L_2(t)RL_{15}(t)$.

$L_3(t)$ – Эффективность

Под эффективностью программного обеспечения (переменная $L_3(t)$) понимается отношение уровня услуг, предоставляемых программным продуктом пользователю при заданных условиях, к объему используемых ресурсов, т.е. эффективность показывает, насколько рационально программа использует ресурсы вычислительной системы (память, процессор) при выполнении своих задач.

На эффективность ($L_3(t)$) влияют такие переменные как:

- Защищенность программного обеспечения (переменная $L_5(t)$), так как эффективность ПО нельзя представить без достаточного уровня защищенности системы при возможных информационных атаках. Справедливо отношение $L_3(t)RL_5(t)$;
- Согласованность системы в целом ($L_6(t)$), так как при некорректной работе взаимосвязных модулей невозможно будет обеспечивать рациональное использование ресурсов системой. Справедливо отношение $L_3(t)RL_6(t)$;
- Завершенность ПО ($L_7(t)$) позволяет обеспечивать эффективность системы, так как весь функционал разрабатываемого ПО будет проверен и корректно использовать вычислительные ресурсы. Справедливо отношение $L_3(t)RL_7(t)$;
- Стабильность программного обеспечения ($L_{10}(t)$) напрямую связана с эффективностью, так как функции системы должны выполняться под разной нагрузкой, и рационально использовать предоставляемые системе ресурсы. Выполняется отношение $L_3(t)RL_{10}(t)$;
- Устойчивость системы к ошибкам ($L_{13}(t)$), как и стабильность, непосредственно связана с эффективностью системы;

- Восстанавливаемость системы ($L_{14}(t)$), связана с эффективностью системы, поскольку возможность системы работать корректно после сбоя обеспечивает рациональное потребление вычислительных ресурсов. Для данной зависимости справедливо отношение $L_3(t)RL_{14}(t)$;
- Понятность ($L_{15}(t)$) позволяет обеспечивать высокий уровень услуг ПО при различных условиях эксплуатации.

$L_4(t)$ – Сопровождаемость

Под сопровождаемостью программного обеспечения понимается свойство ПО, позволяющее минимизировать усилия по внесению в него изменений для устранения ошибок или для модификации в соответствии с изменяющимися потребностями пользователей.

На сопровождаемость ($L_4(t)$) влияют такие переменные как:

- Надежность программного обеспечения (переменная $L_1(t)$), так как корректное выполнение функций системы зависит от возможности устранять в системе потенциальные и существующие ошибки. $L_4(t)RL_1(t)$;
- Защищенность программного обеспечения (переменная $L_5(t)$), так как безопасное функционирование основных процессов системы влияет на возможность корректной модификации. Справедливо отношение $L_4(t)RL_5(t)$;
- Согласованность системы в целом ($L_6(t)$), так как при некорректной работе взаимосвязных модулей невозможно будет обеспечивать внесение изменений. Справедливо отношение $L_4(t)RL_6(t)$;
- Завершенность ПО ($L_7(t)$) позволяет обеспечивать сопровождаемость системы, так как весь функционал разрабатываемого ПО будет проверен и станет возможным удобно изменять систему при необходимости. Справедливо отношение $L_4(t)RL_7(t)$;
- Анализируемость системы ($L_8(t)$) непосредственно влияет на сопровождаемость, так как для внесения изменений в ПО необходимо

достаточно точно анализировать основные функции системы.
 $L_4(t)RL_8(t)$;

- Изменяемость системы ($L_9(t)$) влияет на сопровождаемость обратно пропорционально, поскольку наличие изменений порождает действия по модификации системы;
- Стабильность программного обеспечения ($L_{10}(t)$) напрямую связана с сопровождаемостью, так как функции системы должны выполняться под разной нагрузкой и это не должно приводить к ошибкам модификаций. Выполняется отношение $L_4(t)RL_{10}(t)$;
- Тестируемость программного обеспечения ($L_{11}(t)$) связана с сопровождаемостью, так как возможность проверки модулей системы связана с простотой внесения изменений и их дальнейшей проверкой. Выполняется отношение $L_4(t)RL_{11}(t)$;
- Восстанавливаемость системы ($L_{14}(t)$), связана с сопровождаемостью системы, поскольку возможность системы работать корректно после сбоя обеспечивает возможность внесения модификаций для удовлетворения пожеланий пользователей при самых разных внешних обстоятельствах. Для данной зависимости справедливо отношение $L_4(t)RL_{14}(t)$;
- Понятность системы ($L_{15}(t)$) позволяет обеспечивать высокий уровень внесения модификаций, следовательно, этот критерий влияет на изменяемость ПО.

$L_5(t)$ – Защищенность

Под защищенностью ПО понимается свойство при котором обеспечивается выполнение изделием, реализующим информационную технологию, предписанных функций без нарушений безопасности обрабатываемой информации.

На защищенность ($L_5(t)$) влияют такие переменные как:

- Надежность программного обеспечения (переменная $L_1(t)$), так как корректное выполнение функций системы зависит от безопасного выполнения этих функций. $L_5(t)RL_1(t)$;
- Согласованность системы в целом ($L_6(t)$), так как при некорректной работе взаимосвязанных модулей невозможно будет обеспечивать безопасное выполнение функций системы. Справедливо отношение $L_5(t)RL_6(t)$;
- Завершенность ПО ($L_7(t)$) позволяет обеспечивать защищенность системы, так как весь функционал разрабатываемого ПО будет проверен и, соответственно, количество ошибок безопасности сведется к минимуму. Справедливо отношение $L_5(t)RL_7(t)$;
- Анализируемость системы ($L_8(t)$) непосредственно влияет на защищенность, так как для обеспечения безопасности ПО необходимо достаточно точно анализировать основные функции системы. $L_5(t)RL_8(t)$;
- Изменяемость системы ($L_9(t)$) влияет на сопровождаемость обратно пропорционально, поскольку наличие изменений может порождать ошибки безопасности;
- Стабильность программного обеспечения ($L_{10}(t)$) напрямую связана с защищенностью, так как функции системы должны выполняться под разной нагрузкой и это не должно приводить к ошибкам безопасности. Выполняется отношение $L_5(t)RL_{10}(t)$;
- Тестируемость программного обеспечения ($L_{11}(t)$) связана с защищенностью, так как возможность проверки модулей системы связана с проверкой безопасности системы. Выполняется отношение $L_5(t)RL_{11}(t)$;

- Устойчивость системы к ошибкам ($L_{13}(t)$) связана с безопасностью, так как ошибки безопасности являются подмножеством всех ошибок, возникающих при разработке, сопровождении ИС. $L_5(t)RL_{13}(t)$.

$L_6(t)$ – Согласованность системы в целом

Под согласованностью системы понимают корректную работу всех элементов системы управления и аппарата управления. Отсутствие внутренней противоречивости при работе элементов системы.

На согласованность системы в целом ($L_6(t)$) влияют такие переменные как:

- Надежность программного обеспечения (переменная $L_1(t)$), так как корректное выполнение функций системы влияет на отсутствие противоречивости в работе элементов системы. $L_6(t)RL_1(t)$;
- Эффективность программного обеспечения (переменная $L_3(t)$), так как оптимальное взаимодействие и рациональное использование вычислительных ресурсов порождает целостное состояние системы. Справедливо отношение $L_6(t)RL_3(t)$;
- Сопровождаемость программного обеспечения ($L_4(t)$), так как минимизация времени устранения ошибок обеспечивает консистентное состояние системы в целом. $L_6(t)RL_4(t)$;
- Завершенность ПО ($L_7(t)$) позволяет обеспечивать согласованность системы в целом, так как весь функционал разрабатываемого ПО будет проверен и, соответственно, количество ошибок неконсистентности сведется к минимуму. Справедливо отношение $L_6(t)RL_7(t)$;
- Анализируемость системы ($L_8(t)$) непосредственно влияет на согласованность, так как для обеспечения консистентности модулей ПО необходимо достаточно точно анализировать основные функции системы. $L_6(t)RL_8(t)$;
- Изменяемость системы ($L_9(t)$) влияет на согласованность системы обратно пропорционально, поскольку наличие модификаций может порождать ошибки целостности.

$L_7(t)$ – Завершенность

Под завершенностью понимают совокупность свойств программного средства, характеризующая частоту отказов, обусловленных дефектами программного средства.

На завершенность программного обеспечения ($L_7(t)$) влияют такие переменные как:

- Сопровождаемость программного обеспечения ($L_4(t)$), так как минимизация времени устранения ошибок обеспечивает минимальную частоту отказов системы. $L_7(t)RL_4(t)$;
- Согласованность системы в целом ($L_6(t)$), так как при некорректной работе взаимосвязных модулей невозможно будет обеспечивать минимальную частоту отказов системы. Справедливо отношение $L_7(t)RL_6(t)$;
- Изменяемость системы ($L_9(t)$) влияет на завершенность обратно пропорционально, поскольку наличие модификаций может повышать частоту отказов системы;
- Устойчивость системы к ошибкам ($L_{13}(t)$) влияет на завершенность, поскольку при разных условиях частота отказов системы будет сводиться к минимуму. $L_7(t)RL_{13}(t)$;
- Восстанавливаемость системы ($L_{14}(t)$), связана с завершенностью системы, поскольку возможность системы работать корректно после сбоя обеспечивает минимальную частоту отказов ПО. Для данной зависимости справедливо отношение $L_7(t)RL_{14}(t)$.

$L_8(t)$ – Анализируемость

Анализируемость программного обеспечения подразумевает под собой совокупность свойств программного средства, характеризующая усилия, необходимые для выявления недостатков программного средства или причин его отказов, либо для установления частей, которые должны быть видоизменены.

На анализируемость ($L_8(t)$) влияют такие переменные как:

- Сопровождаемость программного обеспечения (переменная $L_4(t)$), так как для поддержки и внесения изменений в существующее ПО необходимо точно анализировать основные функции системы $L_8(t)RL_4(t)$;
- Защищенность программного обеспечения (переменная $L_5(t)$), поскольку чем проще и быстрее анализировать систему, тем легче находить в ней потенциальные уязвимости и сложнее организовать различные меры защиты от них $L_8(t)RL_5(t)$;
- Согласованность системы в целом (переменная $L_6(t)$) так как при корректной работе взаимосвязных модулей намного проще будет понимание их функций и функций всей системы в целом $L_8(t)RL_6(t)$;
- Завершенность ПО (переменная $L_7(t)$), позволяет оценить весь функционал системы, поскольку ПО, которое больше не будет дополняться и изменяться намного проще анализировать $L_8(t)RL_7(t)$;
- Изменяемость системы (переменная $L_9(t)$). Очевидно, что чем больше система имеет возможностей к изменению и расширению, тем труднее осуществить полный анализ её функционала, поскольку многие части могут быть недоработанными, или же иметь дополнительные функции, являющиеся заделами для последующих наработок $L_8(t)RL_9(t)$;
- Тестируемость системы (переменная $L_{10}(t)$), чем проще провести анализ ПО, тем проще составить для него различные тестовые задачи и проверить взаимодействие модулей, или же их отдельный функционал $L_8(t)RL_{10}(t)$;
- Устойчивость к ошибкам (переменная $L_{11}(t)$), если система легко анализируется и понятны все взаимодействия внутри ее функционала, то становится проще предугадать места возможного появления ошибок, а также выявить причину уже произошедших сбоев $L_8(t)RL_{11}(t)$;

- Понятность системы (переменная $L_{15}(t)$), чем проще и понятнее система, тем меньше трудозатраты на анализ её функционала $L_8(t)RL_{15}(t)$.

$L_9(t)$ – Изменяемость

Степень простоты эффективной и рациональной адаптации для отличающихся или усовершенствованных аппаратных средств, программного обеспечения, других операционных сред или условий использования.

- Эффективность программного обеспечения (переменная $L_3(t)$), чем меньше требования к железу и размеры ПО на физических носителях, тем проще оно изменяется под новые требования $L_9(t)RL_3(t)$;
- Сопровождаемость программного обеспечения (переменная $L_4(t)$), чем больше в ПО возможностей для модификации и адаптивных модулей, тем сложнее их сопровождение из-за повышений требований к навыкам специалистов $L_9(t)RL_4(t)$;
- Защищенность программного обеспечения (переменная $L_5(t)$), чем больше в системе мест для потенциального изменения, тем сложнее найти уязвимость в её модулях $L_9(t)RL_5(t)$;
- Согласованность системы в целом (переменная $L_6(t)$), система, которую легко изменить и перестроить должна обладать высоким уровнем согласованности компонентов $L_9(t)RL_6(t)$;
- Завершенность ПО ($L_7(t)$) если система полностью завершена и согласованна, то в ней не остается легко изменяемых мест и потенциала для реализации дополнительного функционала $L_9(t)RL_7(t)$;
- Анализируемость системы ($L_8(t)$) непосредственно влияет на изменяемость, поскольку увеличения числа адаптивных модулей и усложнение их функционала серьезно затрудняет быстрый анализ данного ПО $L_9(t)RL_8(t)$;
- Стабильность программного обеспечения ($L_{10}(t)$) ухудшается при хорошей изменяемости, поскольку становится сложно предугадать и

отладить все возможные доработки и изменения в функционале ее модулей $L_9(t)RL_{10}(t)$.

$L_{10}(t)$ – Стабильность

Стабильностью характеризуются такие атрибуты программного обеспечения, которые влияют на частоту отказов при ошибках в программном обеспечении.

- Надежность программного обеспечения (переменная $L_1(t)$), так как корректное выполнение функций системы снижает частоту возникновения ошибок в системе. $L_{10}(t)RL_1(t)$;
- Согласованность системы в целом ($L_6(t)$), так как при корректной работе взаимосвязанных модулей снижается частота возникновения ошибок в системе $L_{10}(t)RL_6(t)$;
- Изменяемость системы (переменная $L_9(t)$). Очевидно, что чем больше система имеет возможностей к изменению и расширению, тем больше в ней потенциальных уязвимостей, увеличивающих вероятность возникновения ошибок $L_{10}(t)RL_9(t)$;
- Тестируемость системы (переменная $L_9(t)$), чем выше стабильность системы, тем меньше времени нужно потратить на её тестирование, поскольку частота возникновения ошибок уменьшается $L_{10}(t)RL_{11}(t)$;
- Простота установки (переменная $L_{12}(t)$), меньше ошибок на разных этапах (в т.ч. при установке) – проще установить и подготовить ПО к работе $L_{10}(t)RL_{12}(t)$;
- Устойчивость к ошибкам (переменная $L_{13}(t)$), если система устойчива к ошибкам, то уменьшается частота их возникновения, что соответствует определению стабильности $L_{10}(t)RL_{13}(t)$;
- Восстанавливаемость системы (переменная $L_{14}(t)$), снижение частоты возникновения ошибок позволяет проще восстановить систему после различных сбоев $L_{10}(t)RL_{14}(t)$.

$L_{11}(t)$ – Тестируемость

Степень, до которой могут быть запланированы объективность и реализуемость тестирования, целью которого является проверка соответствия программного обеспечения установленным требованиям.

- Сопровождаемость программного обеспечения (переменная $L_4(t)$), чем проще разобраться в функционале готового ПО и внести в него изменения, тем проще протестировать ПО на соответствие предъявляемым требованиям $L_{11}(t)RL_4(t)$;
- Защищенность программного обеспечения (переменная $L_5(t)$), чем меньше в системе открытых и незашифрованных частей и больше различных проверок, тем сложнее тестирование её функционала $L_{11}(t)RL_5(t)$;
- Согласованность системы в целом ($L_6(t)$), если в системе понятны все взаимосвязи и функции отдельных модулей, то протестировать её становится намного проще $L_{11}(t)RL_6(t)$;
- Завершенность ПО ($L_7(t)$) чем ближе система к завершению, тем меньше возможностей и непроверенных модулей для тестирования остается в её функционале $L_{11}(t)RL_7(t)$;
- Анализируемость системы ($L_8(t)$) практически является синонимом тестируемости, чем проще проанализировать систему, тем проще создать для нее правильные сценарии тестирования ПО $L_{11}(t)RL_8(t)$;
- Стабильность программного обеспечения ($L_{10}(t)$) уменьшение частоты возникновения ошибок уменьшает и количество тестов, необходимых для их выявления, что упрощает тестируемость $L_{11}(t)RL_{10}(t)$;
- Устойчивость к ошибкам (переменная $L_{13}(t)$), устойчивость системы к ошибкам также упрощает тестирование и сокращает время на полную проверку соответствия системы заявленным требованиям $L_{11}(t)RL_{13}(t)$.

$L_{12}(t)$ – Простота установки

Под простотой установки понимают требуемые усилия или ресурсы, необходимые для корректной работы программного средства с учетом нагрузки на него, а также с учетом возможных критических ситуаций.

На простоту установки программного обеспечения ($L_{12}(t)$) влияют такие переменные как:

- Практичность программного обеспечения ($L_2(t)$), поскольку развертывание системы является подпроцессом непосредственной эксплуатации. $L_{12}(t)RL_2(t)$;
- Эффективность ($L_3(t)$), так как чем проще установка системы, тем меньше функционала она будет выполнять и будет менее эффективной;
- Сопровождаемость ($L_4(t)$), так как чем проще установить систему, тем легче ее сопровождать и дорабатывать. Справедливо отношение $L_{12}(t)RL_4(t)$;
- Изменяемость ($L_9(t)$), поскольку чем проще установить программное обеспечение, тем легче его изменять. Справедливо отношение $L_{12}(t)RL_9(t)$.

$L_{13}(t)$ – Устойчивость к ошибкам

Под устойчивостью к ошибкам понимают насколько программное обеспечение будет стабильно работать при возникновении каких-либо внутренних или внешних ошибок, а также насколько быстро и эффективно они будут решаться.

На устойчивость к ошибкам в программном обеспечении ($L_{13}(t)$) влияют такие переменные как:

- Надежность программного обеспечения ($L_1(t)$) напрямую связана с устойчивостью к ошибкам. $L_{13}(t)RL_1(t)$;

- Эффективность ($L_3(t)$), так как, чем система эффективнее, тем более она устойчива к ошибкам. $L_{13}(t)RL_3(t)$;
- Сопровождаемость ($L_4(t)$), ибо сопровождать систему, устойчивую к ошибкам намного проще и легче. $L_{13}(t)RL_4(t)$;
- Защищенность ($L_5(t)$), так как, чем более система будет защищена, тем более она будет устойчива к ошибкам. $L_{13}(t)RL_5(t)$;
- Согласованность системы в целом ($L_6(t)$), так как система более согласована, когда к устойчива к разного рода ошибкам. $L_{13}(t)RL_6(t)$;
- Стабильность ($L_{10}(t)$), ибо чем более система устойчива к ошибкам, тем стабильнее она работает. $L_{13}(t)RL_{10}(t)$;
- Понятность ($L_{14}(t)$), поскольку система более понятна, когда устойчива к ошибкам. $L_{13}(t)RL_{14}(t)$.

$L_{14}(t)$ – Восстанавливаемость

Под восстанавливаемостью понимают способность системы продолжить или в краткий срок восстановить работоспособность при различных критических ситуациях, связанных как с непосредственно железом, так и с ошибками самого программного обеспечения.

На восстанавливаемость программного обеспечения ($L_{14}(t)$) влияют такие переменные как:

- Надежность системы ($L_1(t)$) связана с восстанавливаемостью системы, поскольку возможность системы работать корректно после сбоя влияет на выполнение функций системы. Для данной зависимости справедливо отношение $L_{14}(t)RL_1(t)$;
- Завершенность системы ($L_7(t)$) зависит от способности системы восстанавливаться после отказов. Соотношение $L_{14}(t)RL_7(t)$;

- Стабильность ($L_{10}(t)$), поскольку восстанавливаемость является подмножеством стабильной работы. Выполняется отношение $L_{14}(t)RL_{10}(t)$.
- Устойчивость к ошибкам ($L_{13}(t)$), так как систему устойчивую к ошибкам намного проще восстановить. Выполняется отношение $L_{14}(t)RL_{13}(t)$.

$L_{15}(t)$ – Понятность

Под понятностью понимают способность поддерживать систему и возможность ее улучшать и развивать, затрачивая при этом малое количество ресурсов.

На понятность программного обеспечения ($L_{15}(t)$) влияют такие переменные как:

- Практичность системы ($L_2(t)$), непосредственно связана с ее понятностью. Для данной зависимости справедливо отношение $L_{15}(t)RL_2(t)$;
- Эффективность ($L_3(t)$) позволяет обеспечивать высокий уровень услуг ПО при различных условиях эксплуатации, особенно если система при этом понятна. Выполняется отношение $L_{15}(t)RL_3(t)$;
- Сопровождаемость системы ($L_4(t)$) позволяет дорабатывать программное обеспечение и развивать его, чем более понятна система, тем легче ее сопровождать и наоборот. Выполняется отношение $L_{15}(t)RL_4(t)$;
- Согласованность системы в целом ($L_6(t)$), система более согласована, когда проста и понятна. $L_{15}(t)RL_6(t)$;
- Анализируемость ($L_8(t)$), систему намного легче анализировать, когда она понятна. Выполняется отношение $L_{15}(t)RL_8(t)$;

- Изменяемость ($L_9(t)$), чем понятнее система, тем легче ее изменять. Выполняется отношение $L_{15}(t)RL_9(t)$;
- Тестируемость ($L_{11}(t)$), чем проще и понятнее система, тем легче ее тестировать. Выполняется отношение $L_{15}(t)RL_{11}(t)$.