

Тестирование в ЖЦ ПО

Водопадная модель

V-образная модель

Итерационная инкрементальная модель

Спиральная модель

Гибкая модель

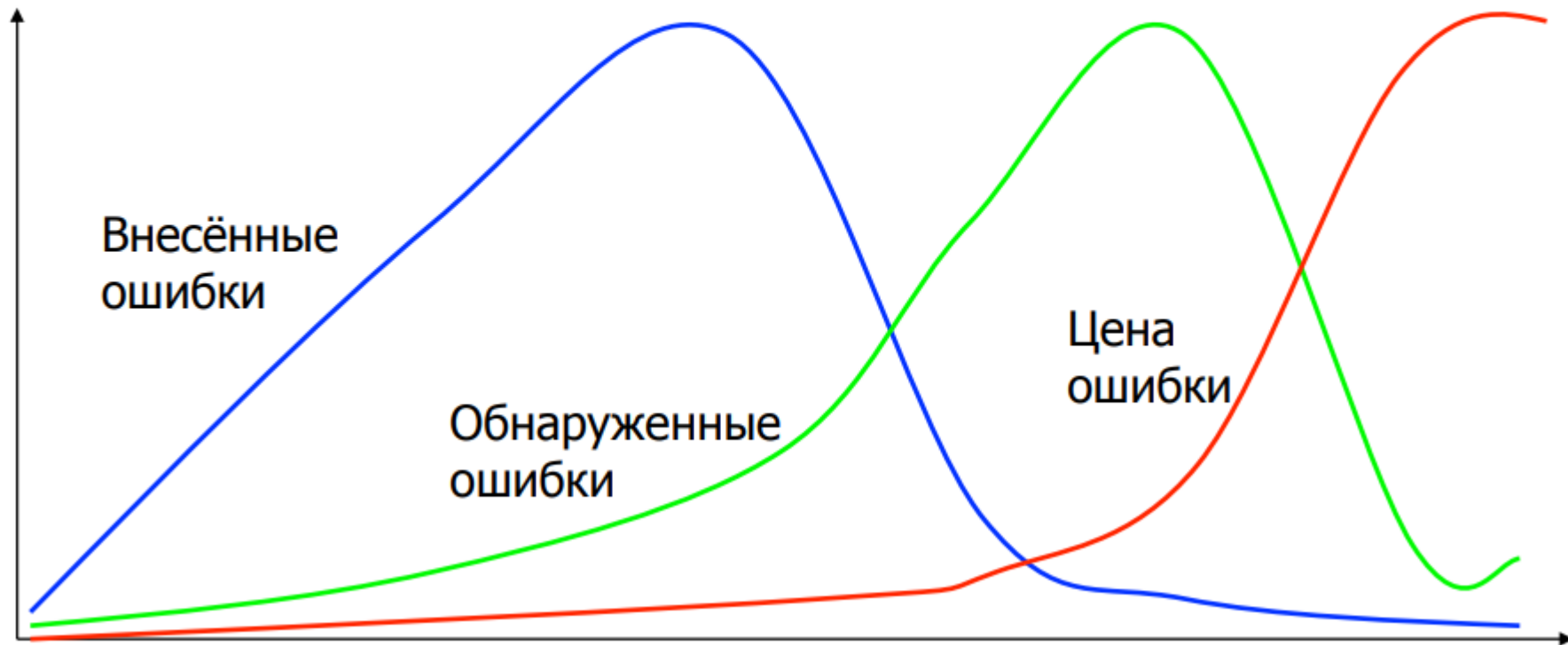
Сравнение

Водо- падная модель

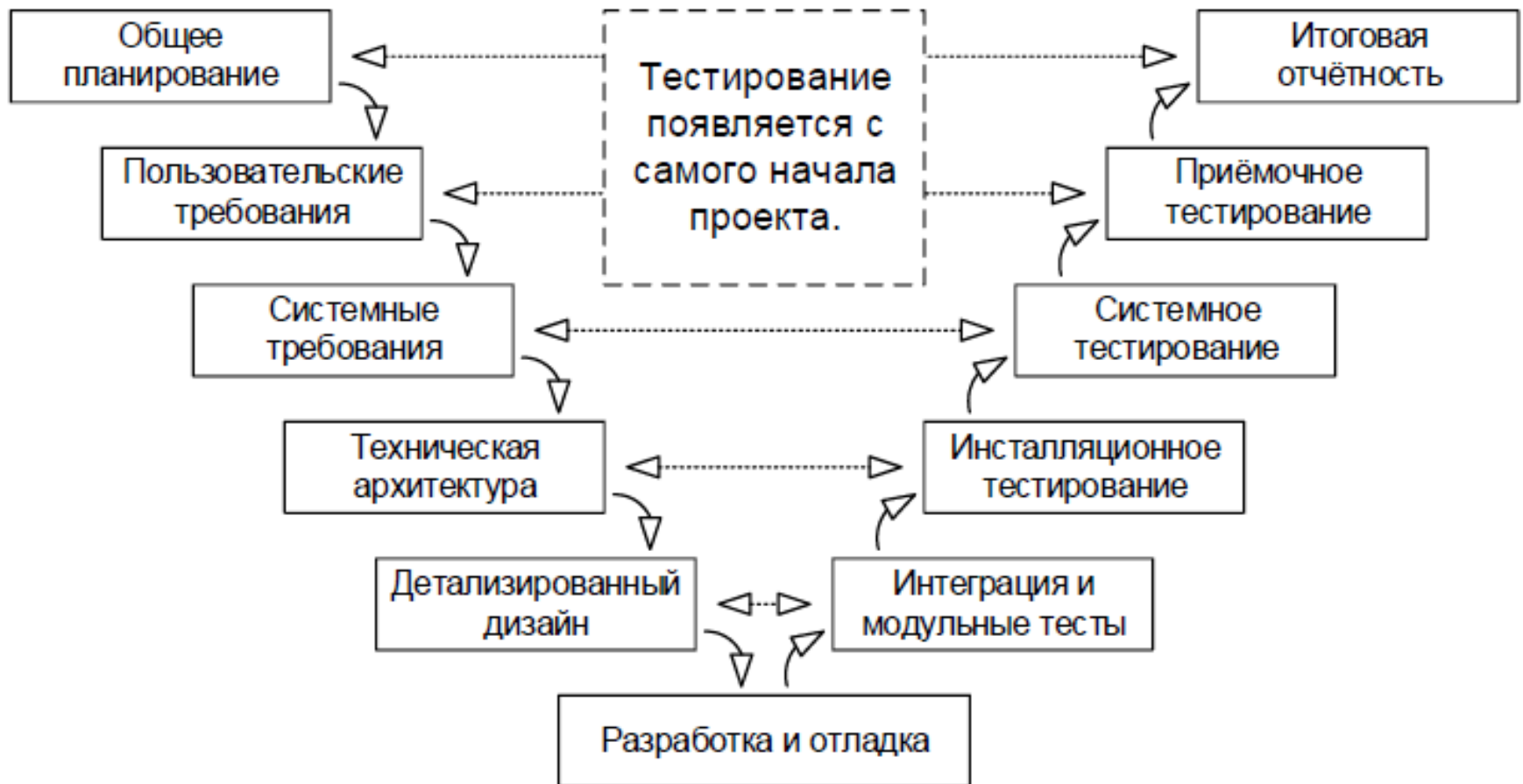


Цена ошибки

Анализ	Проектирование	Реализация	Тестирование		Эксплуатация
			Unit	System	



V-образная модель



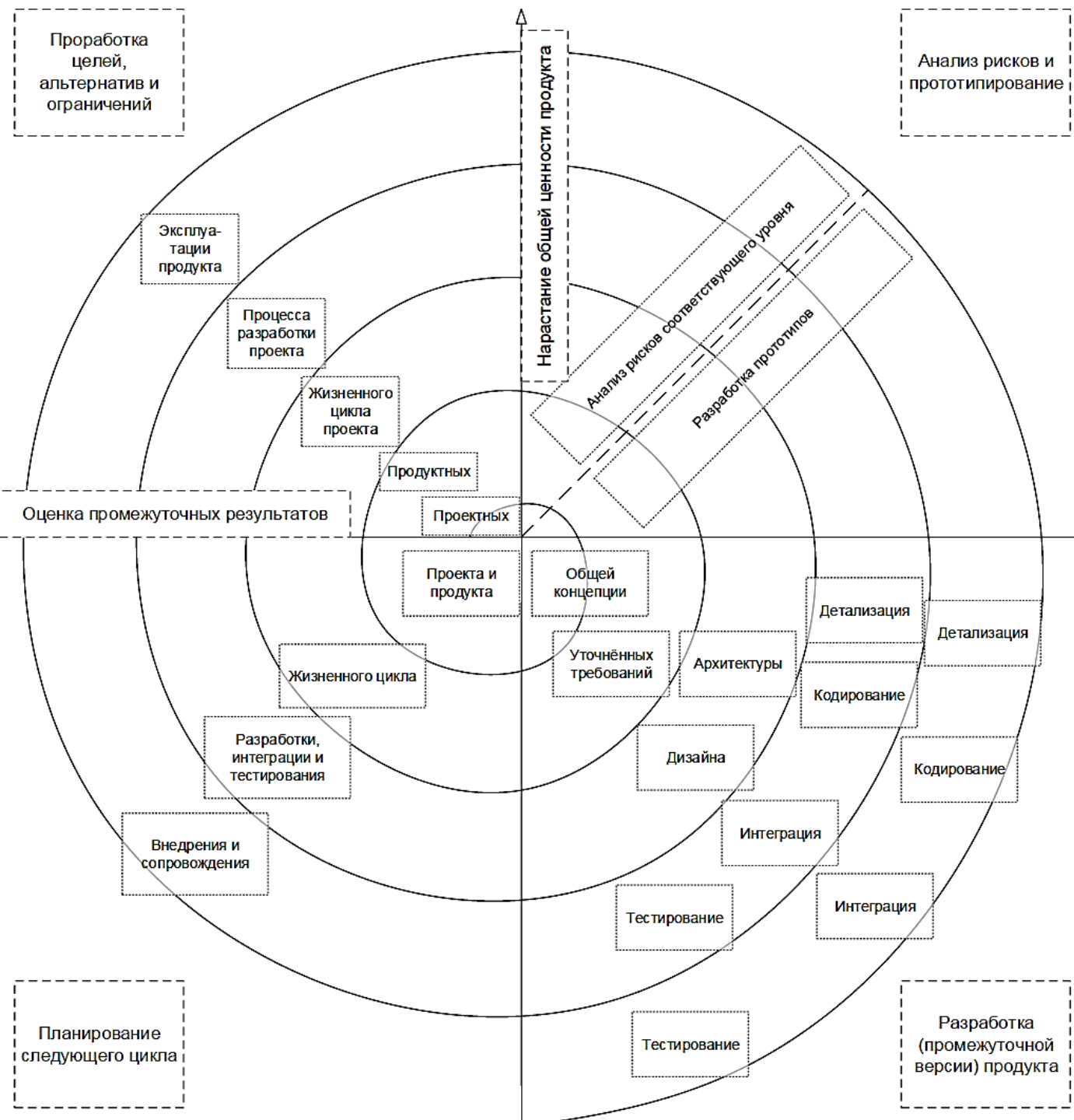
- упрощённо можно сказать, что на каждой стадии «на спуске» нужно думать о том, что и как будет происходить на соответствующей стадии «на подъёме». Тестирование здесь появляется уже на самых ранних стадиях развития проекта, что позволяет минимизировать риски, а также обнаружить и устранить множество потенциальных проблем

Итерационная инкрементальная модель



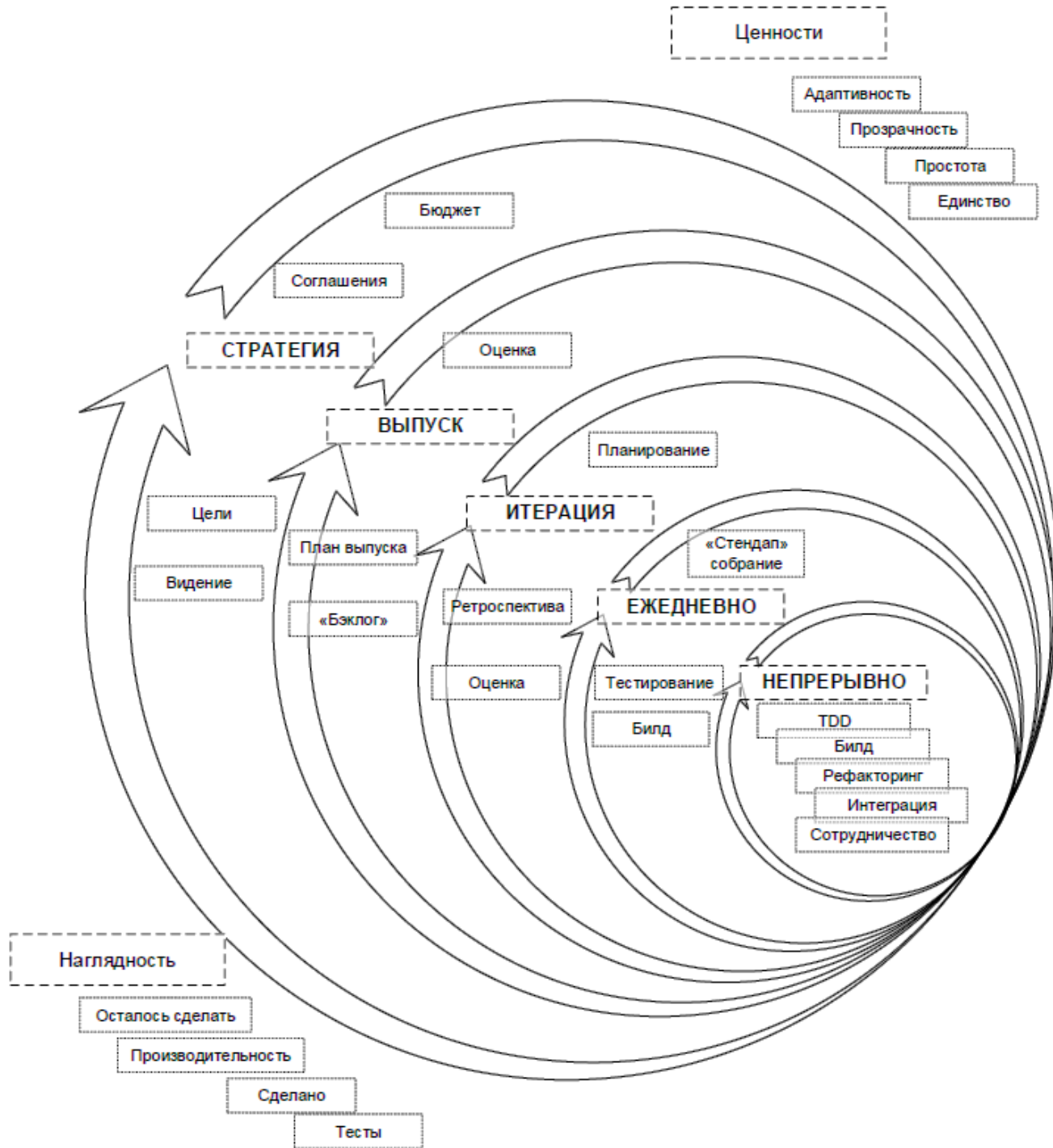
- Разбиение проекта на относительно небольшие промежутки (итерации), каждый из которых в общем случае может включать в себя все классические стадии, присущие водопадной и v-образной моделям. Итогом итерации является приращение (инкремент) функциональности продукта, выраженное в промежуточной сборке

Спиральная модель

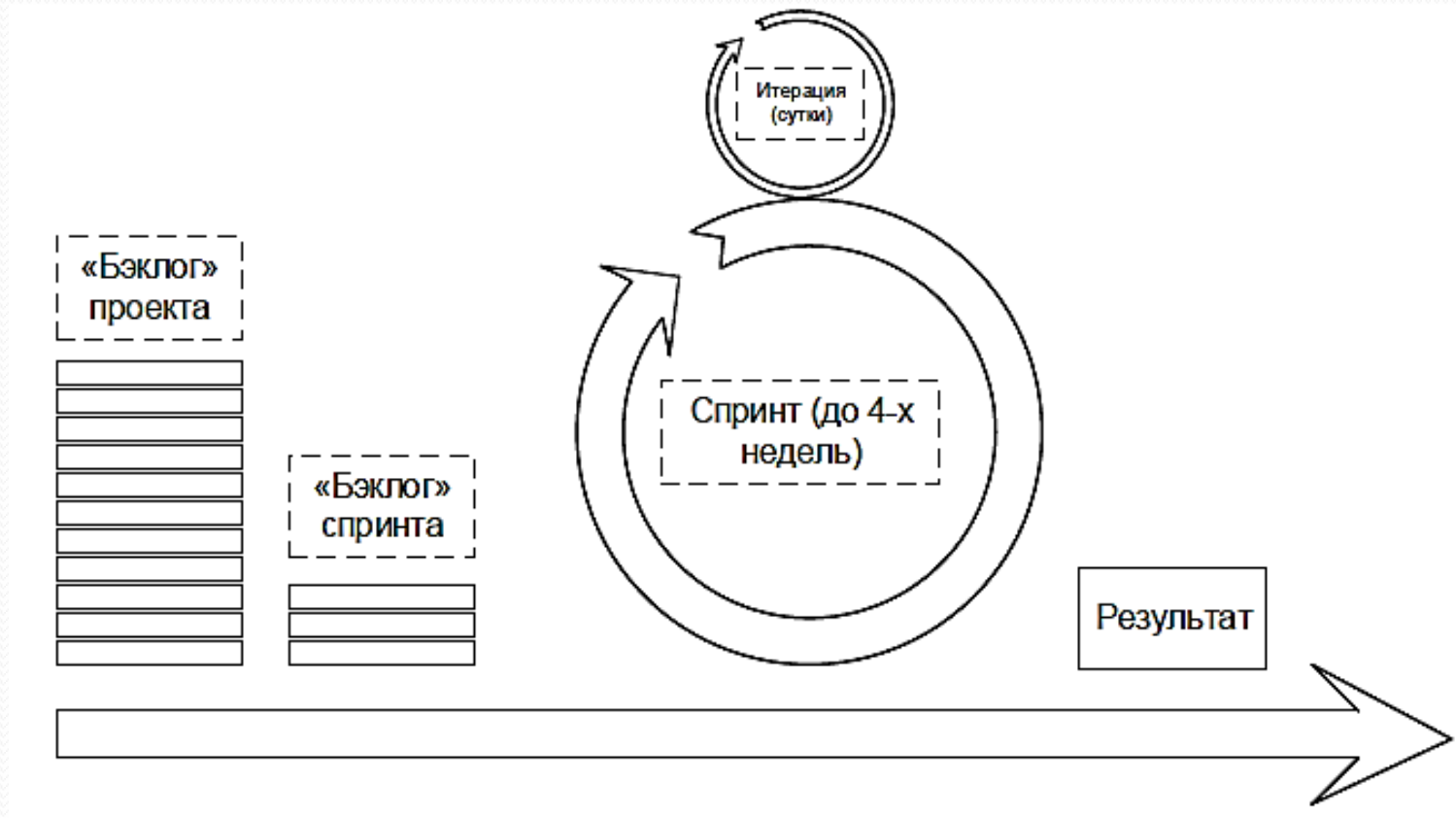


- повышенное внимание рискам - преимущество для разработки концептуальных проектов, в которых требования являются сложными и нестабильными (могут многократно меняться по ходу выполнения проекта).

Гибкая модель



Гибкая модель (итерация)



- впервые был достигнут осязаемый результат в снижении бюрократической составляющей и максимальной адаптации процесса разработки ПО к мгновенным изменениям рынка и требований заказчика

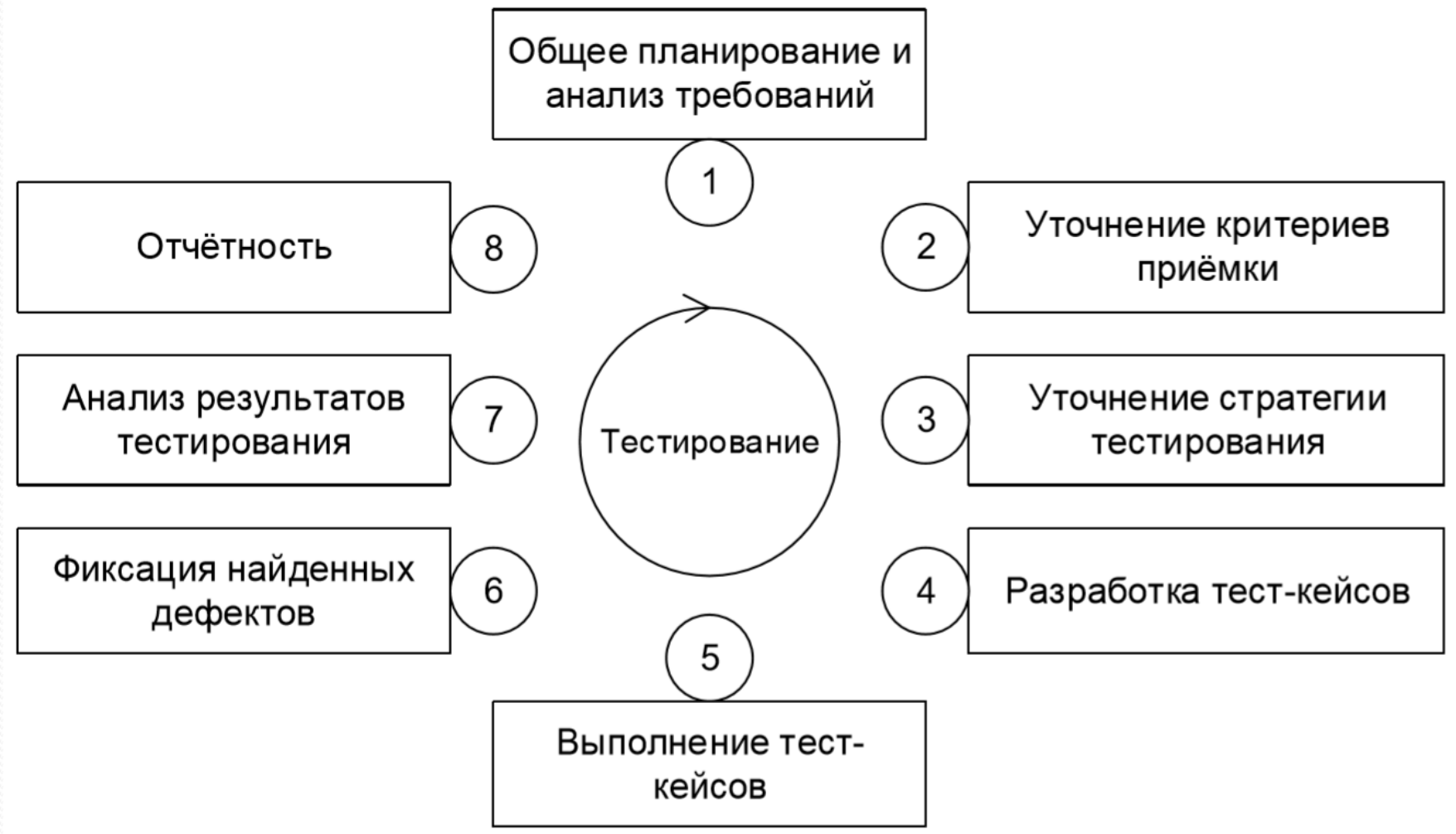
Сравнение моделей разработки ПО

Модель	Преимущества	Недостатки	Тестирование
Водопадная	<ul style="list-style-type: none">• У каждой стадии есть чёткий проверяемый результат.• В каждый момент времени команда выполняет один вид работы.• Хорошо работает для небольших задач.	<ul style="list-style-type: none">• Полная неспособность адаптировать проект к изменениям в требованиях.• Крайне позднее создание работающего продукта.	<ul style="list-style-type: none">• С середины проекта.
V-образная	<ul style="list-style-type: none">• У каждой стадии есть чёткий проверяемый результат.• Внимание тестированию уделяется с первой же стадии.• Хорошо работает для проектов со стабильными требованиями.	<ul style="list-style-type: none">• Недостаточная гибкость и адаптируемость.• Отсутствует раннее прототипирование.• Сложность устранения проблем, пропущенных на ранних стадиях развития проекта.	<ul style="list-style-type: none">• На переходах между стадиями.

Сравнение моделей разработки ПО

Итерационная инкрементальная	<ul style="list-style-type: none"> • Достаточно раннее прототипирование. • Простота управления итерациями. • Декомпозиция проекта на управляемые итерации. 	<ul style="list-style-type: none"> • Недостаточная гибкость внутри итераций. • Сложность устранения проблем, пропущенных на ранних стадиях развития проекта. 	<ul style="list-style-type: none"> • В определённые моменты итераций. • Повторное тестирование (после доработки) уже проверенного ранее.
Спиральная	<ul style="list-style-type: none"> • Глубокий анализ рисков. • Подходит для крупных проектов. • Достаточно раннее прототипирование. 	<ul style="list-style-type: none"> • Высокие накладные расходы. • Сложность применения для небольших проектов. • Высокая зависимость успеха от качества анализа рисков. 	
Гибкая	<ul style="list-style-type: none"> • Максимальное вовлечение заказчика. • Много работы с требованиями. • Тесная интеграция тестирования и разработки. • Минимизация документации. 	<ul style="list-style-type: none"> • Сложность реализации для больших проектов. • Сложность построения стабильных процессов. 	<ul style="list-style-type: none"> • В определённые моменты итераций и в любой необходимый момент.

Жизненный цикл тестирования



Это не догма, но принципы остаются

Жизненный цикл тестирования

- Стадия 1 (общее планирование и анализ требований) дает ответ на вопросы:
 - что нам предстоит тестировать;
 - как много будет работы;
 - какие есть сложности;
 - всё ли необходимое у нас есть и т.п.
- Стадия 2 (уточнение критериев приёмки) позволяет сформулировать или уточнить метрики и признаки возможности или необходимости
 - начала тестирования (entry criteria),
 - приостановки (suspension criteria)
 - возобновления (resumption criteria) тестирования,
 - завершения или прекращения тестирования (exit criteria).
- Стадия 3 (уточнение стратегии тестирования): рассматриваются и уточняются те части стратегии тестирования (test strategy), которые актуальны для текущей итерации.

Жизненный цикл тестирования

- Стадия 4 (разработка тест-кейсов) посвящена разработке, пересмотру, уточнению, доработке, переработке и прочим действиям с тест-кейсами, наборами тесткейсов, тестовыми сценариями и иными артефактами, которые будут использоваться при непосредственном выполнении тестирования.
- Стадия 5 (выполнение тест-кейсов) и стадия 6 (фиксация найденных дефектов) тесно связаны между собой и фактически выполняются параллельно: дефекты фиксируются сразу по факту их обнаружения в процессе выполнения тест-кейсов.
 - Однако зачастую после выполнения всех тест-кейсов и написания всех отчётов о найденных дефектах проводится явно выделенная стадия уточнения, на которой все отчёты о дефектах рассматриваются повторно с целью формирования единого понимания проблемы и уточнения таких характеристик дефекта, как важность и срочность.
- Стадия 7 (анализ результатов тестирования) и стадия 8 (отчётность) также тесно связаны между собой и выполняются практически параллельно. Формулируемые на стадии анализа результатов выводы напрямую зависят от плана тестирования, критериев приёмки и уточнённой стратегии, полученных на стадиях 1, 2 и 3.
- Полученные выводы оформляются на стадии 8 и служат основой для стадий 1, 2 и 3 следующей итерации тестирования. Таким образом, цикл замыкается.