

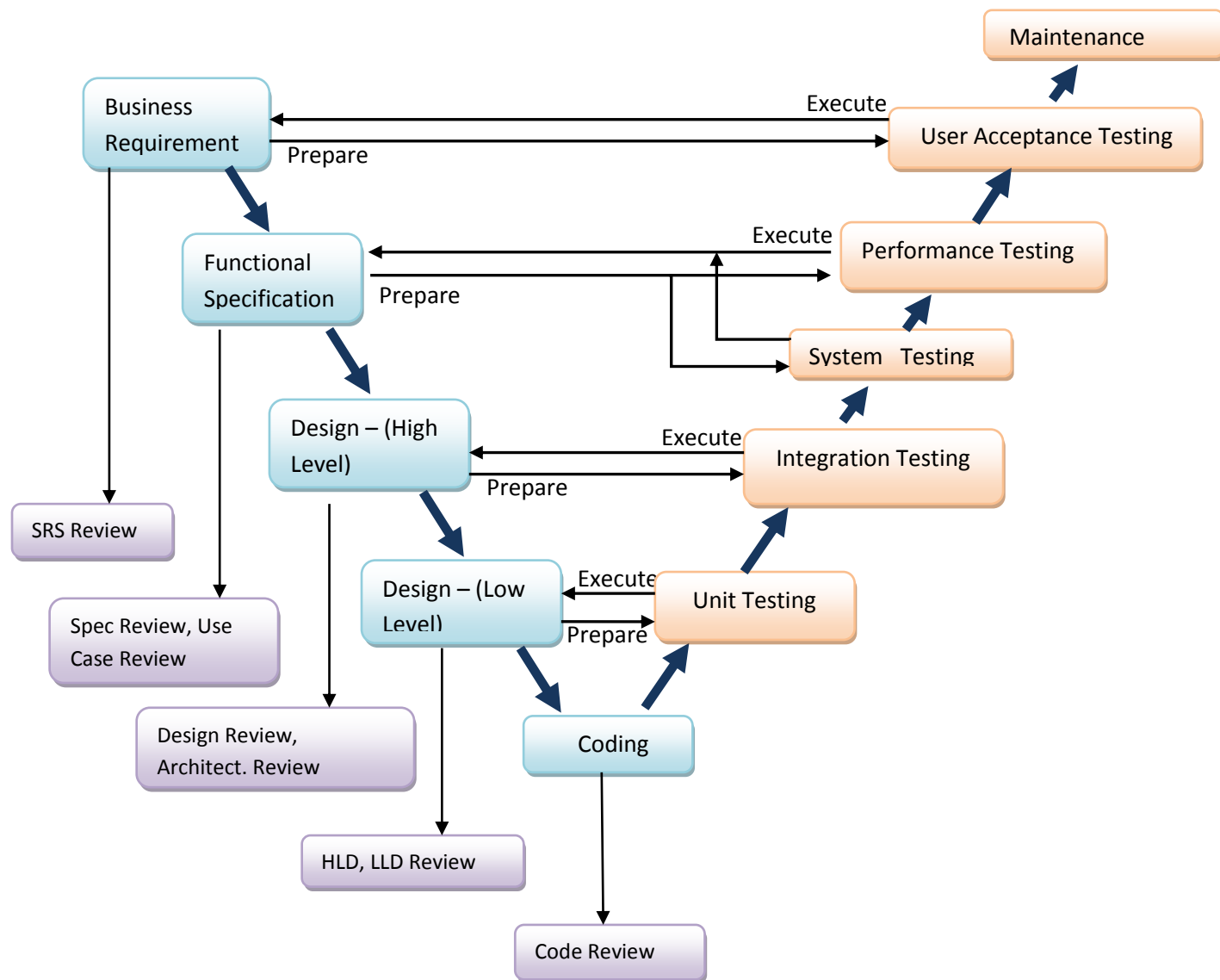
Agile Test Methodology Vs V Model

The object of the article is to uncover the disadvantages of the traditional V model of testing over the Agile test methodology. In doing so we will cover the following points:

- What is V model? Advantages and disadvantages of V model.
- What is Agile Testing? Advantages and disadvantages of Agile Testing.
- Why Agile is preferred over V model?

V model

V-model is a software development model that is based on the relationships between each phase of the development life cycle as described in a typical Waterfall model of software development and its associated phase of testing



The V-model involves building a logical V shape sequence where the testing techniques associated with the design are reflected as descending and are applied for the “verification” and connected to the requirements or specifications parts are reflected as ascending and are applied for “validation”. Equal weight to coding and testing in the V-model

gives software development process. The V-model ordains that the code testing documentation is written in tandem with the development phases that means, for instance, the integration tests should be documented as and when the high level design is finalized and the unit tests should be ready as and when the detailed specifications are laid down.

The idea of the V-model is to have a implementation plan for the software testing at each level namely component, interface, system, acceptance and release of the software project which need to be adhered to eliminate discrepancies in the software simultaneously rather than waiting for the software development process to complete before handling it to the software testing professionals.

The V-Model is a Software life cycle model.

It explains how you go from high level concept to released product.

- At the top of the left V line, are the High level concepts.
- At the bottom of the V is the coding phase.
- At the top right V line is the released product.

The reason for calling it a V is because most of the phases on the left have a corresponding phase of activity on the right.

Advantages of V Model

- It is also called as verification and validation Model.
- This means the verification and validation will be done side by side.
- It emphasis the strict process flow to develop a quality product
- The errors occurred in any phase will be corrected in that phase itself.

Disadvantages of V Model

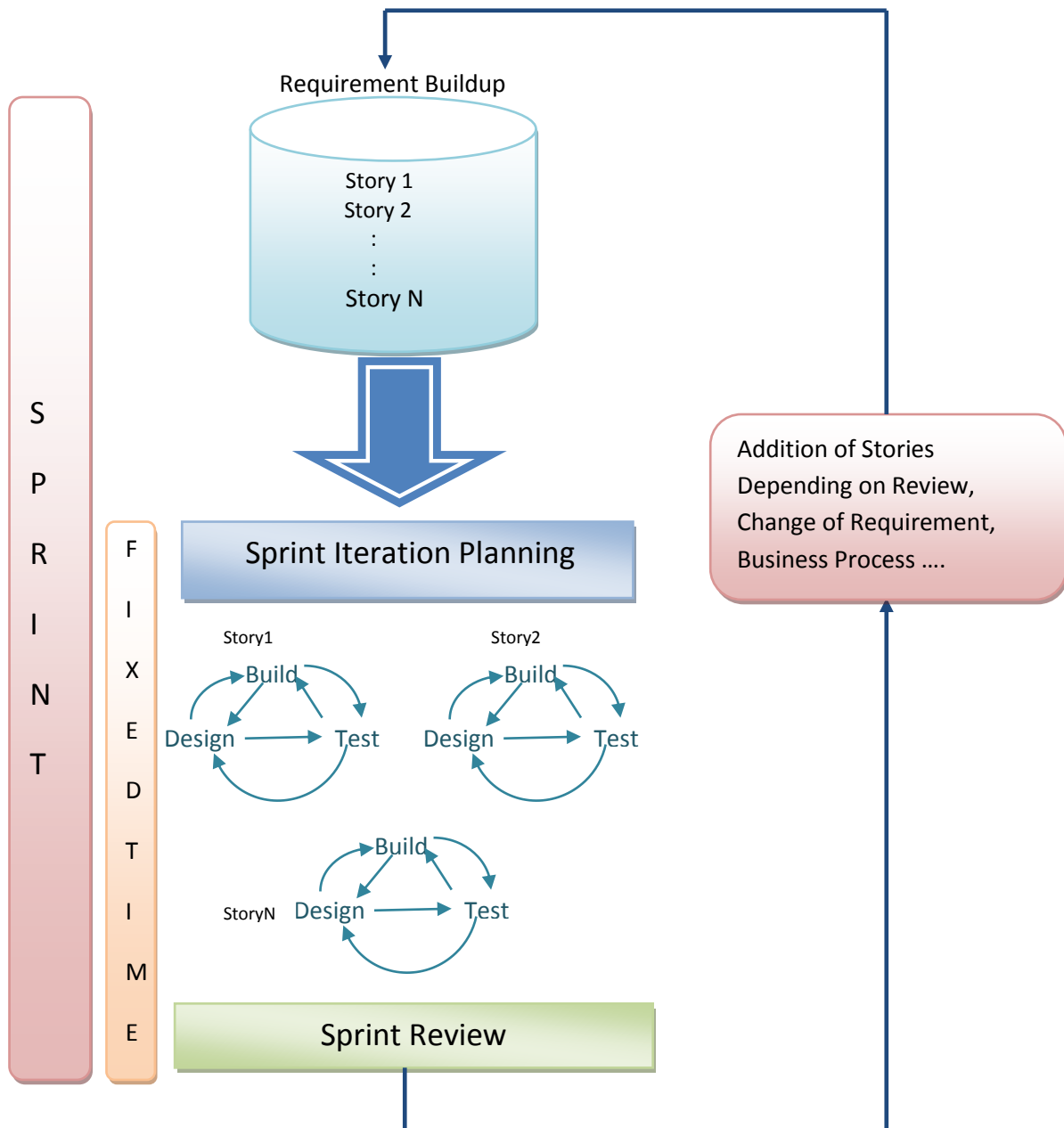
- The biggest disadvantage of V-model is that it's very rigid and the least flexible. If any changes happen mid way, not only the requirements documents but also the test documentation needs to be updated.
- The V-Model addresses software development within a project rather than a whole organization.
- The organization and execution of operation, maintenance, repair and disposal of the system are not covered by the V-Model. However, planning and preparation of a concept for these tasks are regulated in the V-Model.

Agile Testing

Agile testing is a software testing practice that follows the rules of the agile manifesto, treating software development as the customer of testing. Agile testing involves testing from the customer perspective as early as possible, testing early and often as code becomes available and stable enough from module/unit level testing.

Agile methods were developed as a response to the issues that the traditional V-Model and waterfall methodologies had with defining requirements and delivering a product that turned out to be not what the end user actually wanted and needed.

The concept of “the team being responsible for quality” i.e. “the whole team concept” and not just the testing team, is a key value of agile methods.



Testing from the beginning of the start of the project and continually testing throughout the project lifecycle, is the foundation on which agile testing is built. Every practice, technique or method is focused on this one clear goal

Agile methodologies are designed to break the software down into manageable parts that can be delivered earlier to the customer. The aim of any Agile project is to deliver a basic working product as quickly as possible and then to go through a process of continual improvement. An Agile project is characterized by having a large number of short delivery cycles (sprints) and priority is given to the feedback-loops from one cycle to the next. The feedback-loops drive continuous improvement and allow the issues that inevitably occur (including the way requirements have been defined or the quality of the solution provided) to be dealt with much earlier in the development life cycle. To achieve this, companies need to re-think their approach to delivery and have their previously independent contributors (Business Analysts, Developers, Testers, End Users etc.) worked together in teams.

The key challenges for a tester on an agile project are:

- No traditional style business requirements or functional specification documents. We have small documents (story cards developed from the 4×4 inch cards) which only detail one feature. Any additional details about the feature are captured via collaborative meetings and discussions.
- You will be testing as early as practical and continuously throughout the lifecycle so expect that the code won't be complete and is probably still being written
- Your acceptance Test cases are part of the requirements analysis process as you are developing them before the software is developed
- The development team has a responsibility to create automated unit tests which can be run against the code every time a build is performed
- With multiple code deliveries during the iteration, your regression testing requirements have now significantly increased and without test automation support, your ability to maintain a consistent level of regression coverage will significantly decrease

The role of a tester in an Agile project requires a wider variety of skills:

- Domain knowledge about the system under test
- The ability to understanding the technology be used
- A level of technical competency to be able to interact effective with the development team.

Advantages of Agile Testing

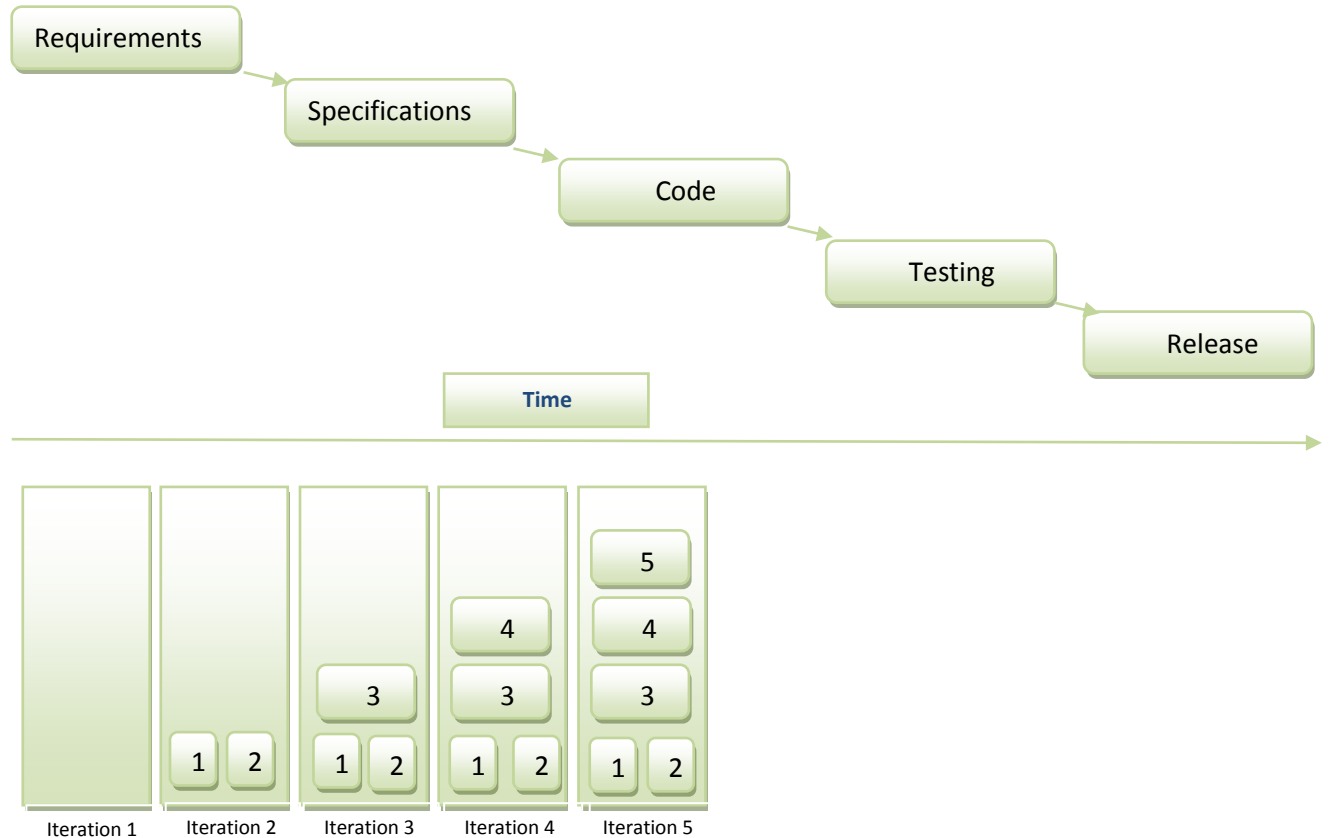
- Agile methodology drastically reduces testing cycle time as the testing is done parallel to the development activities.
- The iterative nature of **agile development** means features are delivered incrementally, enabling some benefits to be realized early as the product continues to develop
- A key principle of **agile development** is that testing is integrated throughout the lifecycle, enabling regular inspection of the working product as it develops. This allows the product owner to make adjustments if necessary and gives the product team early sight of any quality issues.
- Small incremental releases made visible to the product owner and product team through its development help to identify any issues early and make it easier to respond to change.

Disadvantages of Agile Testing

- Requirements are clarified just in time for development and can be documented in much less detail due to the timeliness of conversations. However this can mean less information available to new starters in the team about features and how they should work. It can also create potential misunderstandings if the teamwork and communication aren't at their best and difficulties for team members (especially testers) that are used to everything being defined up front.
- Requirements emerge and evolve throughout the development. This creates the very meaning of agile - flexibility. Flexibility to change course as needed and to ensure delivery of the right product. There are two big flip sides to this principle though. One is the potential for **scope creep**, which we all know can create the risk of **ever-lasting projects**. The other is that there is **much less predictability**, at the start of the project and during, **about what the project is actually going to deliver**. This can make it harder to define a business case for the project, and harder to negotiate fixed price projects. Without the maturity of a strong and clear vision, and the discipline of fixing timescales and trading scope, this is potentially very dangerous.

Why Agile is preferred over V model?

Despite of above disadvantages Agile is preferred over V model these days. The main reasons behind these are explained below.



In the phased approach diagram, it is clear that testing happens at the end, right before release. The diagram is idealistic; because it gives the impression there is as much time for testing as there is for coding. In many projects, this is not the case. The testing gets “squished” because coding takes longer than expected, and because teams get into a code-and-fix cycle at the end.

Agile is iterative and incremental. This means that the testers test each increment of coding as soon as it is finished. Iteration might be as short as one week, or as long as a month. The team builds and tests a little bit of code, making sure it works correctly, and then moves on to next piece that needs to be built. Programmers never get ahead of the testers, because a story is not “done” until it has been tested.