Agile Software Development (TCS 855)

Unit-V Agile Market

Market Scenario and Adoption of Agile, Roles in Agile Projects

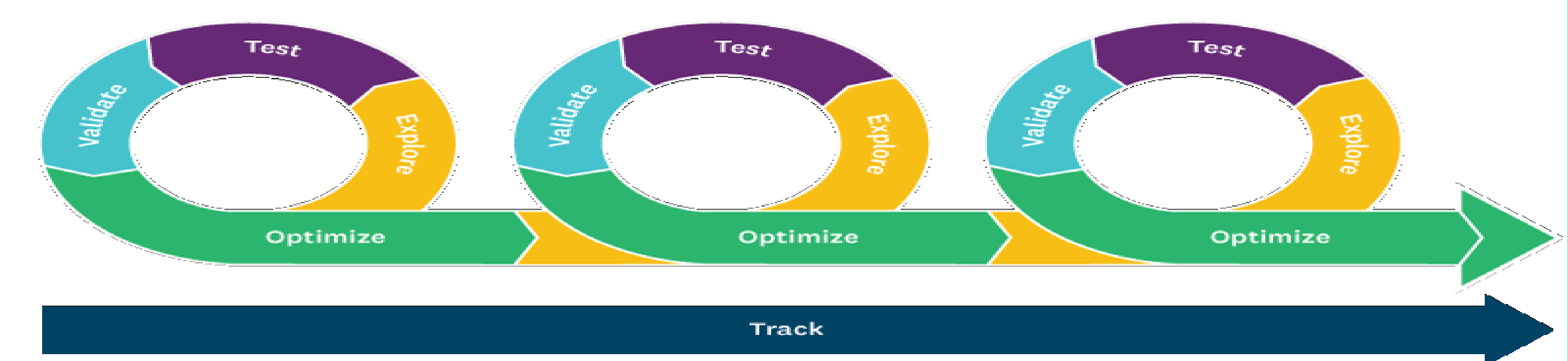
Prof.(Dr.) Santosh Kumar

Department of Computer Science and Engineering

Graphic Era Deemed to be University, Dehradun

Market Scenario and Adaptation of Agile

- Today’s business landscape is changing rapidly, and marketers are often at the forefront when it comes to facing the brunt of a major change.
- Agile market research is revolutionizing how companies get insights today.
- Agile market research is the application of agile software development methods (think iterative development, sprints, and minimum viable products) to the practice of market research, enabling teams and companies to test, iterate, and launch products, concepts, or campaigns faster and more effectively thanks to a more customer-centric approach.



An example of an agile market research framework.

- New products get to the right customers faster, new business ideas find validation before crucial investments are made, companies can keep a real-time pulse on their existing customers and larger addressable market.
- Some great news for research and insights professionals, product and strategy teams, and medium to small companies with few resources and a deep hunger for market data.
- When we combine the agile methodology with the innovative DIY market research tools available today, we get increasingly powerful capabilities that anyone can put to work for a variety of use cases: product development, consumer behavior, concept testing, brand performance, market sizing, business plan validation, and more.

- The emergence of new technologies, channels and trends at shorter intervals has rendered traditional long-term marketing practices irrelevant.
- It is a common experience that by the time marketers gain deep understanding of new and disruptive technologies and master its applications, a faster and better solution has already arrived on the scene.
- List of five ways in which agile market research is revolutionizing how companies big and small get mission-critical business insights today.

1. Market research automation makes us faster and more productive.
2. Artificial intelligence makes us smarter.
3. New products serve increasingly specific use cases
4. Insights experts are not the only ones getting insights
5. Agile research is becoming part of many teams’ annual strategy

Advantages of Agile Marketing

- Agile Marketing ensures optimum efficiency as every member – designer, coder, writer, etc. – contributes towards quickly launching the campaign.
- This eliminates the downtime of any member or resource due the campaign being stuck at any stage in the marketing lifecycle.
- Besides, outsourcing, which is integral the very concept of agility, ensures that each member of the team does what they are best at, while other tasks are outsourced to feasible service providers.
- Agile marketers modify their tactics on the basis of target-audience preferences, thereby maximizing the effectiveness of their effort.
- Leveraging advanced business and marketing intelligence tools, they can track the performance of their campaigns, incorporate best practices and avoid repetition of same mistakes.
- The efficiency and effectiveness of Agile Marketing also ensures significant cost-saving. By preventing wastage of time and resources, marketers can actually tide over poorly performing, even a failed campaign, as they can easily optimize and iterate the entire cycle for less time and cost, unlike traditional marketing practices.

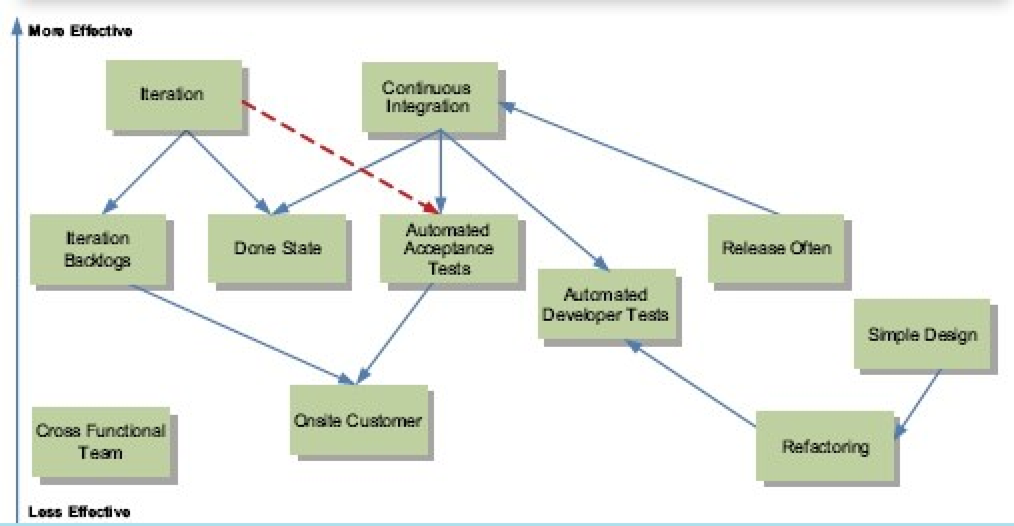
Adoption of Agile

- There are a numerous of Agile practices out there. Which ones are right for you and your team?
- What are the business values you want out of adopting Agile and what is your organization’s context?
- This Refcard is focused on helping you evaluate and choose the practices for your team or organization when getting to market faster is of prime importance.
- Instead of focusing on entire methods such as Scrum and XP, we will talk about the practices that are the building blocks of these methods such as iterations and automated developer tests. We will answer two basic questions:

  i. What Agile practices should you consider to improve Time to Market?

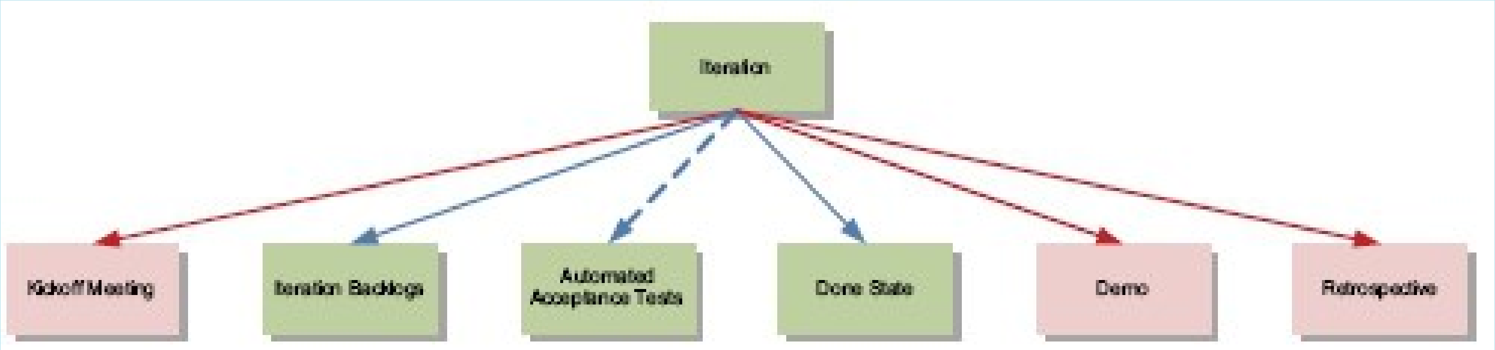
  ii. How should you go about choosing from those practices given your organization and context?

Agile Practices Improve Time to Market



- These are the Agile practices that improve time to market.
- The most effective practices are near the top of the diagram.
- Therefore iteration is more effective than Onsite Customer for improving time to market.
- The arrows indicate dependencies. Continuous Integration depends on Automated Developer Tests for it to be effective.

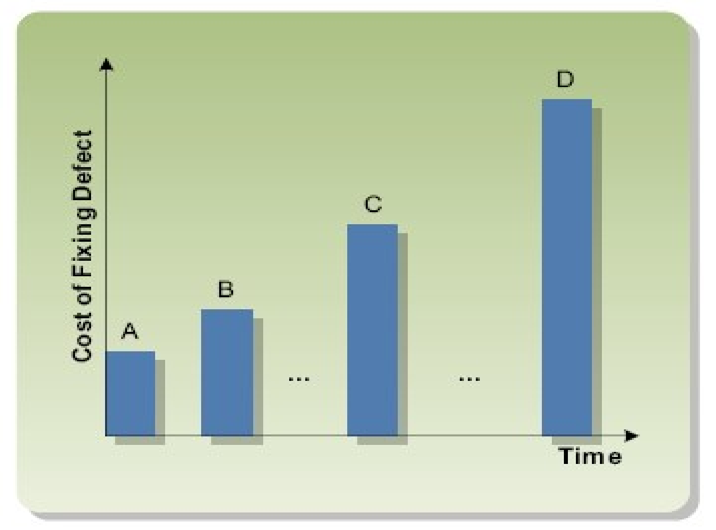
Iteration



- Practices in pink are ones that don’t directly address time to market but are needed to support practices that do (hence a dependency).
- They are not described in this Refcard but can be found in the external references.

Continuous Integration

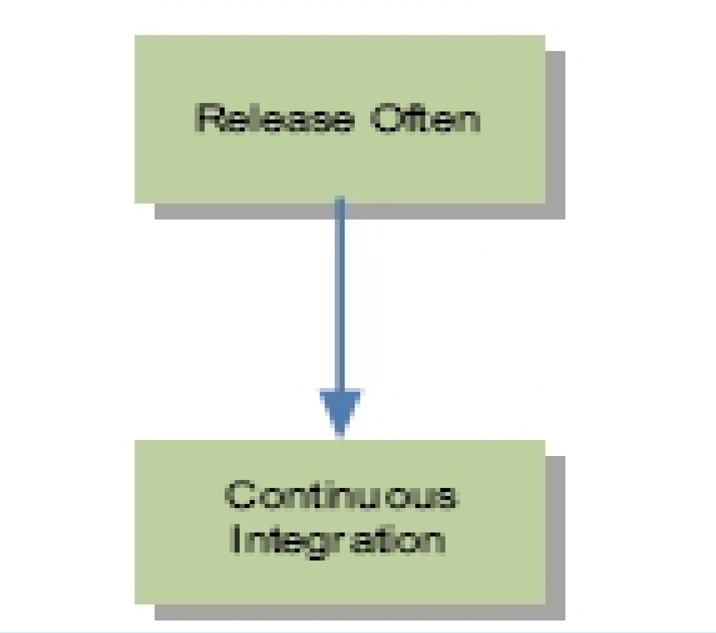
- The cost of fixing a defect increases over time because of context switching, communication, and bugs being built on existing bugs.



Release Often

- Release your software to your end customers as often as you can without inconveniencing them.

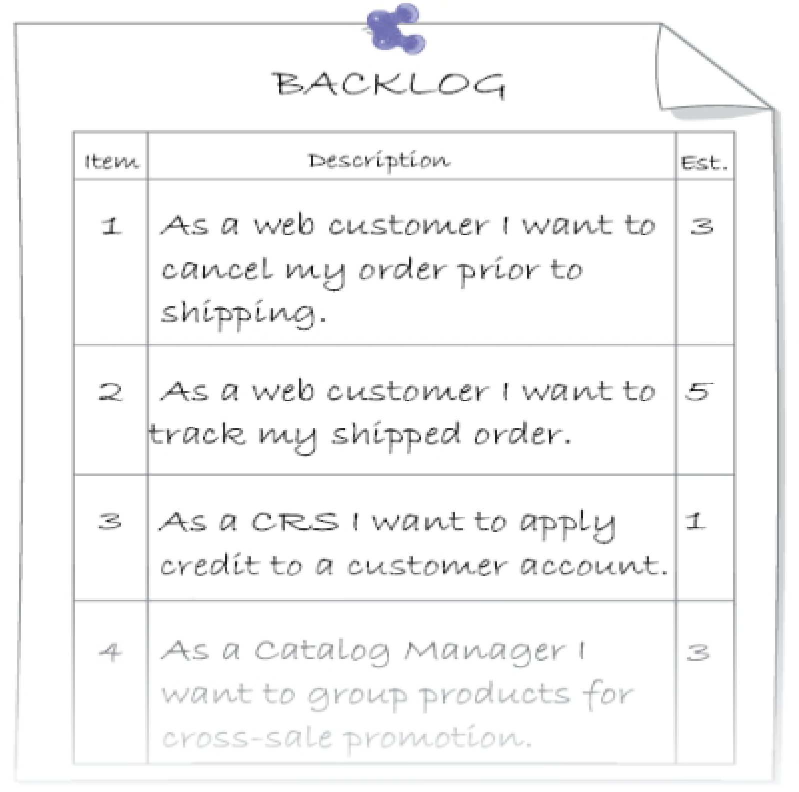
- Releasing often streamlines your development process and makes you deal with the pains of getting software good enough to go live.
- A team that releases often faces the pains and addresses the problems that make deployment difficult so that releasing is just another development task.
- You are on a project where releasing often will enable you to produce revenue earlier.
- Having new features available frequently will not inconvenience your customer base. The quality of your releases is superb and your customers eagerly await your next release (instead of religiously keeping away from your 1.0 releases).


Release Often

Continuous Integration

Iteration Backlog

- A backlog is a prioritized list of requirements. There are two common flavors of backlogs, one for the current iteration and one for the product.
- The product backlog contains all of the requirements prioritized by value to the customer. The iteration backlog is a list of requirements that a team has committed to building for an iteration.
- Properly prioritized backlogs that are used to set the goals for every iteration ensure that the team is always working on the most important requirements.
- When paired with iterations that produce working, tested software, backlogs give a development team the option to release at the end of any iteration having always worked on the most important issues.
- An expert on business value is needed to be part of the team to prioritize the backlog. If your team has such a person or someone that can coordinate with the business stakeholders to do so then use a product backlog.
- If you are using iterations then use an iteration backlog to set clear goals for the iterations and a release backlog to maintain long-term goals.

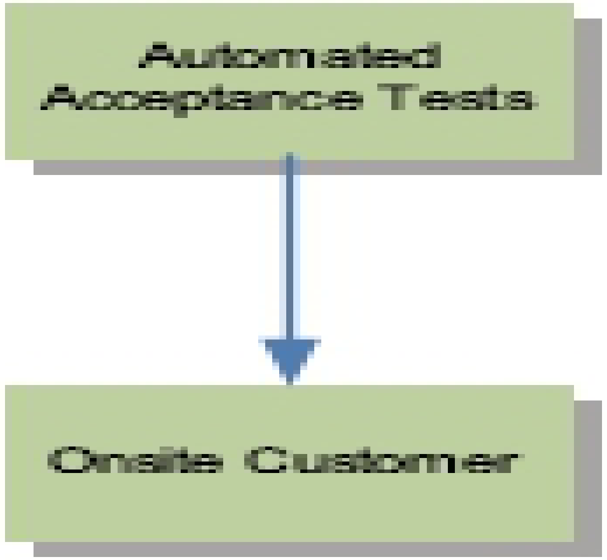


BACKLOG

Item	Description	Est.
1	As a web customer I want to cancel my order prior to shipping.	3
2	As a web customer I want to track my shipped order.	5
3	As a CRS I want to apply credit to a customer account.	1
4	As a Catalog Manager I want to group products for cross-sale promotion.	3

Automated Developer Tests

- Automated developer tests are a set of tests that are written and maintained by developers to reduce the cost of finding and fixing defects—thereby improving code quality—and to enable the change of the design as requirements are addressed incrementally.
- Automated developer tests reduce the time to market by actually reducing the development time. This is accomplished by reducing a developer’s time in debugging loops by catching errors in the safety-net of tests.
- You are on a development team that has decided to adopt iterations and simple design and will need to evolve your design as new requirements are taken into consideration.
- Or you are on a distributed team. The lack of both face-to-face communication and constant feedback is causing an increase in bugs and a slowdown in development.

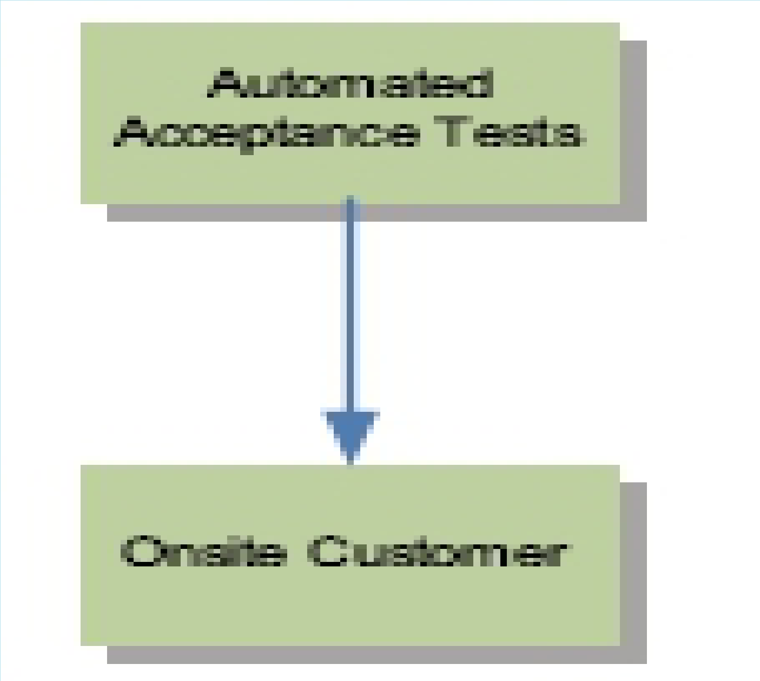


Automated Acceptance Tests

Onsite Customer

Automated Acceptance Tests

- Automated acceptance tests are tests written at the beginning of the iteration that answer the question: “what will this requirement look like when it is done?”.
- This means that you start with failing tests at the beginning of each iteration and a requirement is only done when that test passes.
- This practice builds a regression suite of tests in an incremental manner and catches errors, miscommunications, and ambiguities very early on. This, in turn, reduces the amount of work that is thrown away and enables faster development as you receive early feedback when a requirement is no longer satisfied.

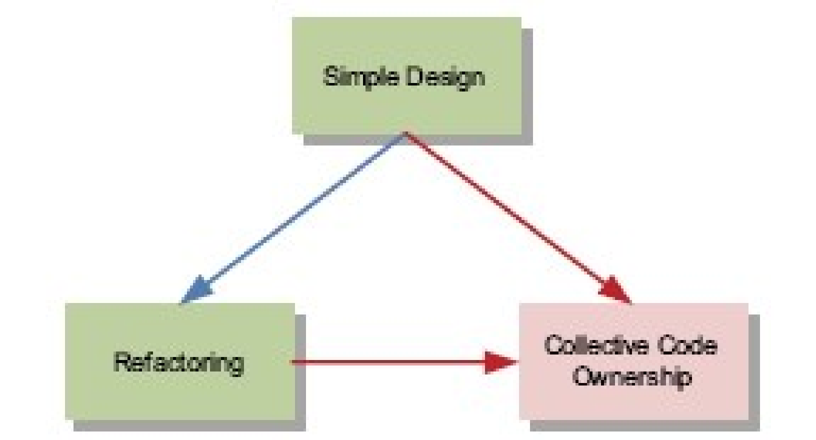


Onsite Customer

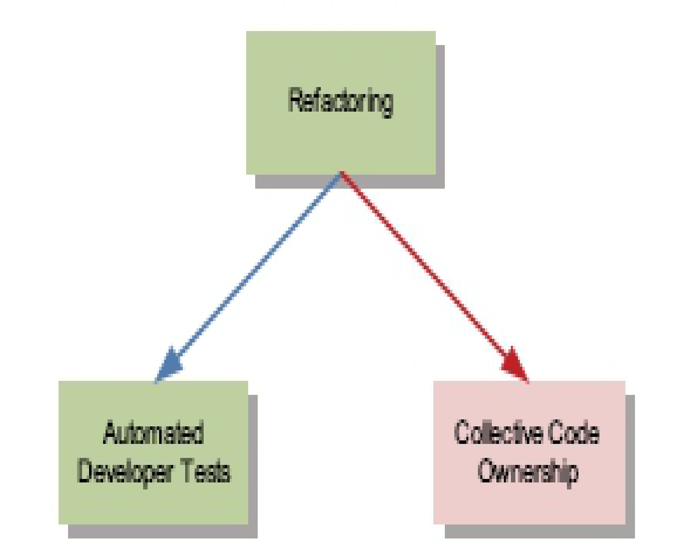
- The onsite customer role in an Agile development team is a representative of the users of the system who understands the business domain of the software.
- The customer owns the backlog, is responsible for writing and clarifying requirements, and responsible for checking that the software meets the requirements specified.
- The role of customer helps improve time to market by supporting the developers by giving them clear requirements, providing clarifications and verifying that the software does really meet the needs of the user base.
- The customer provides early feedback to the development team so they never spend more than an iteration down a blind alley.
- Finally, having a customer who correctly prioritizes a backlog allows the team to deliver the most important items first when time is of the essence.
- The practice of onsite customer works best when the development team can be co-located with one or more domain experts.
- The person fulfilling the customer role is crucial to the success of the team and therefore will need sufficient time and resources to do the job.

Simple Design

- If a decision between coding a design for today's requirements and a general design to accommodate for tomorrow's requirements needs to be made, the former is a simple design.
- Simple design meets the requirements for the current iteration and no more.
- Simple design improves time to market because you build less code to meet the requirements and you maintain less code afterwards.
- Simple designs are easier to build, understand, and maintain.
- Simple design should only be used when your team also is writing automated developer tests and refactoring.
- A simple design is fine as long as you can change it to meet future requirements.



Refactoring



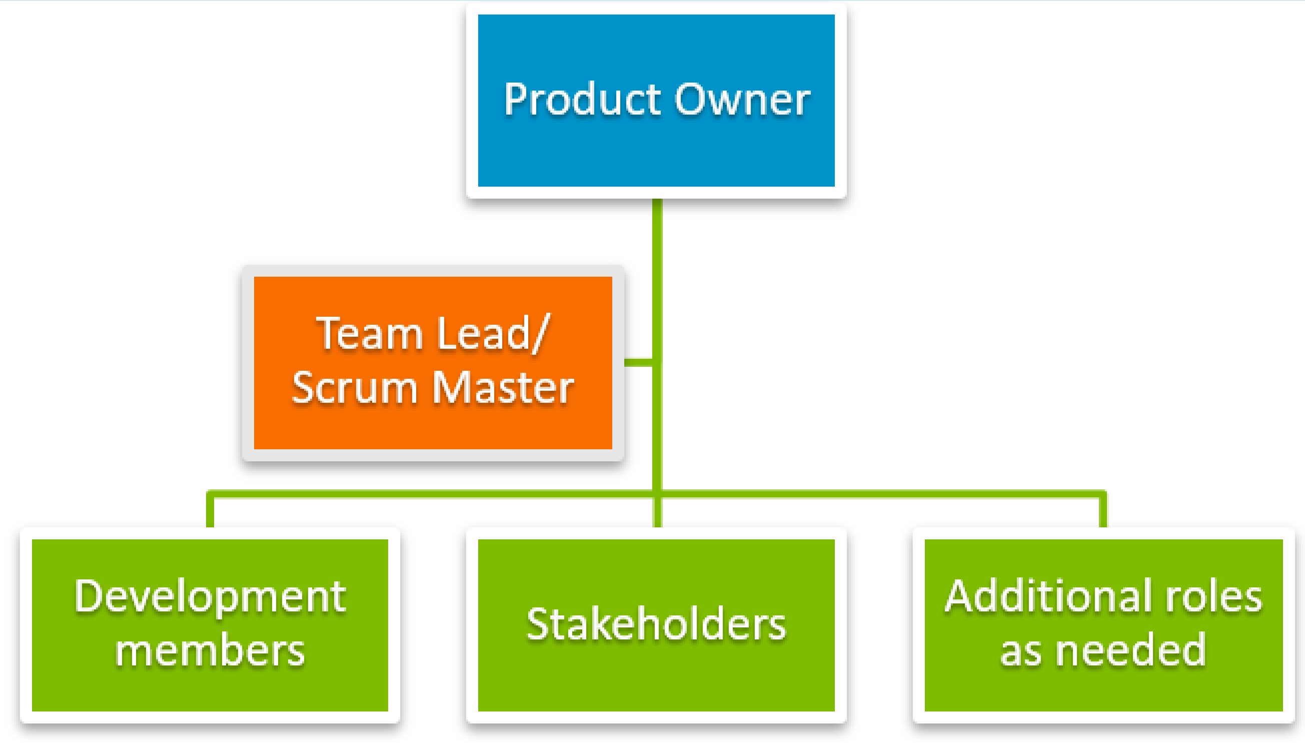
- The practice of Refactoring code changes the structure (i.e., the design) of the code while maintaining its behavior.
- Collective code ownership is needed because a refactoring frequently affects other parts of the system.
- Automated developer tests are needed to verify that the behavior of the system has not changed after the design change introduced by the refactoring.
- Refactoring improves time to market by supporting practices like Simple Design which, in turn, allow you to only write the software for the features that are needed now.
- You are on a development team that is practicing automated developer tests. You are currently working on a requirement that is not well-supported by the current design.

Cross-Functional Team

- The team that utilizes the Cross Functional Team practice is one that has the necessary expertise among its members to take a requirement from its initial concept to a fully deployed and tested piece of software within one iteration.
- A requirement can be taken off of the backlog, elaborated and developed, tested, deployed.
- Cross-functional teams primarily affect time to market by enabling true iterative and incremental development. Resource bottlenecks are resolved and teams can build features end-to-end.
- There is a hardening cycle at the end of each release indicating unresolved integration issues. Building a slice of functionality end-to-end in your system finds errors early and requires diverse expertise of many different people.

Agile Roles & Responsibilities

- The roles and responsibilities within the Scrum framework for Agile implementation.
- Agile teams are often comprised of the following key roles and responsibilities:



Agile Team Roles

1. **Product owner**

   The key responsibilities of a Product Owner include:

   ✓Scrum backlog management

   ✓Release management

   ✓Stakeholder management

2. **Team lead/Scrum master**

   The role may involve:

   ✓Facilitating the daily Scrum and Sprint initiatives

   ✓Communicating between team members about evolving requirements and planning

   ✓Coaching team members on delivering results

   ✓Handling administrative tasks such as conducting meetings, facilitating collaboration, and eliminating hurdles affecting project progress

   ✓Shielding team members from external interferences and distractions

3. **Development team members**

   The required skills might be wrapped up in one or more dev team members:

   ✓ Product designer

   ✓ Writer

   ✓ Programmer

   ✓ Tester

   ✓ UX specialist

4. Stakeholders

The stakeholder may be:

- ✓The end user of the product
- ✓Business executives
- ✓Production support staff
- ✓Investors
- ✓External auditors
- ✓Scrum team members from associated projects and teams

4. Additional roles for larger Scrum projects

These can include:

- ✓**Technical and domain experts** with the knowledge of technology as well as a wide variety of stakeholder requirements or expectations.
- ✓**An independent testing** and audit team may join the Scrum team members and work throughout the product development lifecycle.
- ✓**An Integrator may be required** among large teams that work on independent but closely coordinated subsystems for a project. The responsibility for the Integrator would include integration of the subsystems as well as testing that may be performed by external testing teams.
- ✓**An Architect Owner** may be required for architectural envisioning, planning and decision making.