Mindtree
A Larsen & Toubro Group Company

Whitepaper

A PoV on Development Models

Introduction

Development models define the way projects are structured, the various phases of the project delivery and the roles associated. In this whitepaper, we discuss various development models along with their key characteristics and relevance. By understanding the relevance of the development models, we can adopt the most suitable model for the program.

Assembly Line Development Model

Assembly line development is classic method and is used commonly across the globe. The model uses waterfall execution methodology wherein each project phase flows in a sequential order, with strict dependency on the previous phase.

This model consists of different designated roles for each responsibility. Most of the project activities such as code review and testing is done manually in this model. This model has strict quality gates to ensure the excellence of the software, and relies on heavy integration testing driven manually.

Assembly Line Development Model Design

In order to implement this model, we need to assign a designated role for each of the team members. The roles include comprising front-end and back-end developers, and manual, integration and automation test engineers etc. Each team member is only responsible for his/ her role.

The following are the execution steps to implement the assembly model:

1. Requirement understanding and analysis
2. Architecture and design
3. Development
4. Unit and functional testing
5. Deployment

We have defined the assembly model process in Figure 1

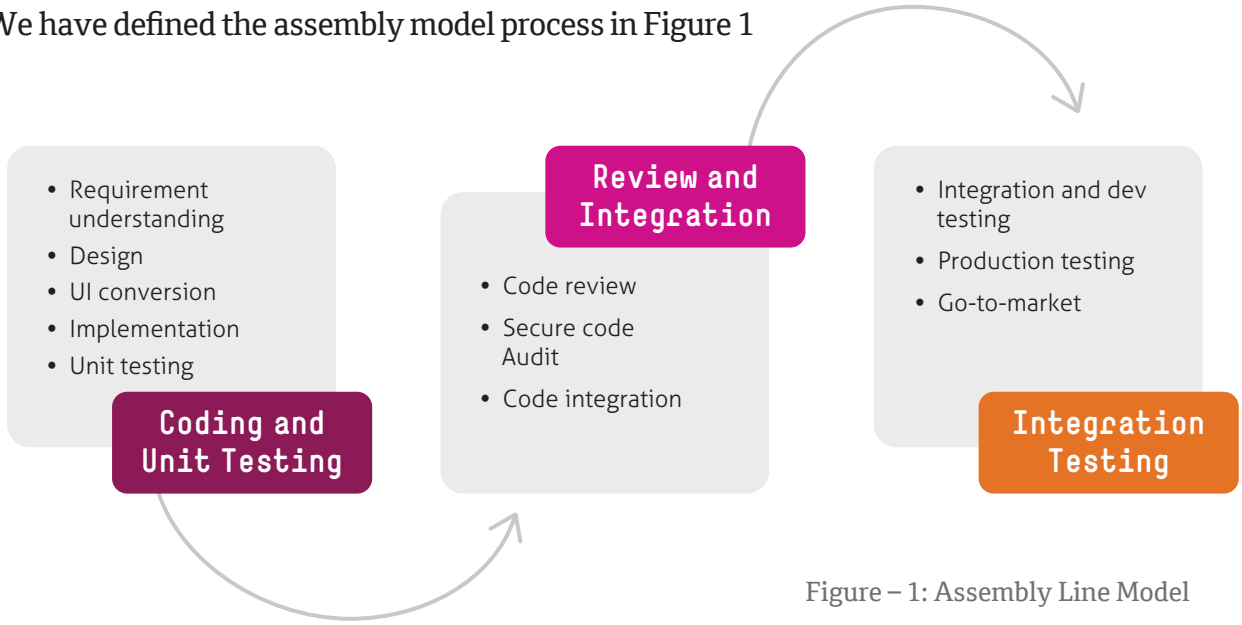


Figure – 1: Assembly Line Model

Lean Supervisory Development Model

Lean supervisory development is an agile model where each team member has a clearly-defined responsibility. In this model, each team member has end-to-end ownership toward sprint goals. Automated tools are used for quality control, testing and release management. A continuous integration/continuous delivery (CI/ CD) pipeline improves the process of development during integration/ testing, delivery and deployment phases. The DevOps pipeline includes all stages like build, test, release, deploy, validation and compliance. We deploy static code analysis tools like SonarQube to scan the code and report quality issues. QA automation for UI and functional testing also gives instant feedback during build.

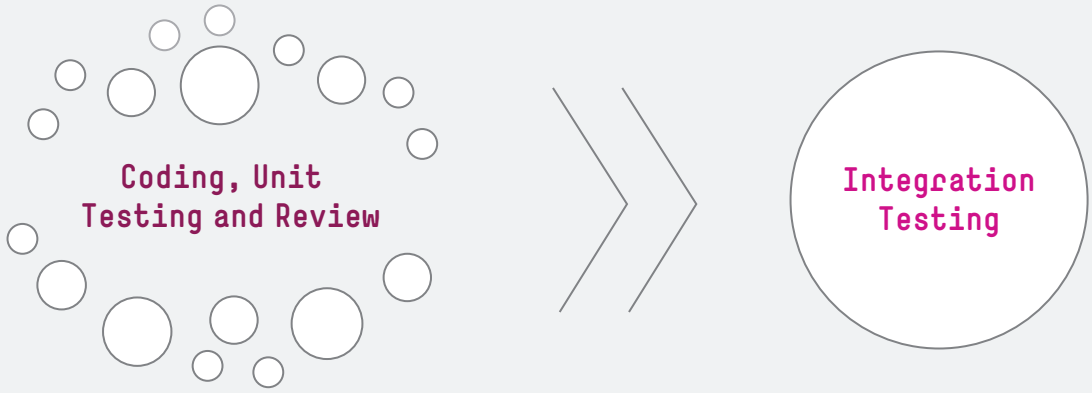
Given below are the main features of the lean supervisory model

- End-to-end ownership with developers
- CI/CD-based automation
- Code scanner for instant feedback
- QA automation

Lean Supervisory Development Model Design

The implementation of a lean supervisory development model is effective only with the right resources and tools. These resources consists of full stack development and this model is best suited with Agile OR Scaled Agile frameworks. All the tools have to be properly investigated as per project need and then executed. Following are the key success factors:

- Groom the features into detailed user stories with enough technical and user/ test scenarios
- Ensure code quality by multiple levels of code review like peer review on task level, architect review on user story level, PO review on feature/ release level.
- Unit, system, integration and automation tests
- Static/ dynamic code analysis by automatic tooling
- Secure code analysis/ audit, pen test



- Disciplined team and end-to-end ownership with scrum team
- User story grooming in detail till task level
- Detailed design
- Unit system, integration and automation tests
- Static/ dynamic code analysis by automatic tooling
- Secure code analysis/ audit, pen testing
- Instant feedback
- External penetration testing
- CI/CD automated pipeline

- Integration and dev testing
- Production testing
- Go-to-market

Figure – 2 Lean Supervisory Model

Modular Development Model

The modular development model is a modern way of development, and is an enhanced version of the lean supervisory model for delivering high quality results. It includes all qualities of the lean supervisory model with additional aspects like test-driven development. The team comprises the full stack developer and SDET (Software Development Engineer in Test). The main driver of this model is end-to-end integrated automation and automation.

Modular Development Model Design

The implementation of a modular development model is slightly complex and time-consuming in the initial stage, when compared to a lean supervisor model. Forming a disciplined team could take sometime, given that the tools need to be integrated properly with CI/ CD, which in turn leads to faster execution. Following are the characteristics to be considered:

- Groom the features into detailed user stories with enough technical and user/ test scenarios
- Test Driven Development (TDD) approach for development
- Ensure code quality by multiple levels of code review like peer review on task level, architect review on user story level, PO review on feature/ release level.
- Unit, system, integration and automation tests
- Static/ dynamic code analysis by automatic tools such as SonarQube, FindBugs
- Secure code analysis/ audit, pen testing

We have depicted the modular development model in Figure 3.

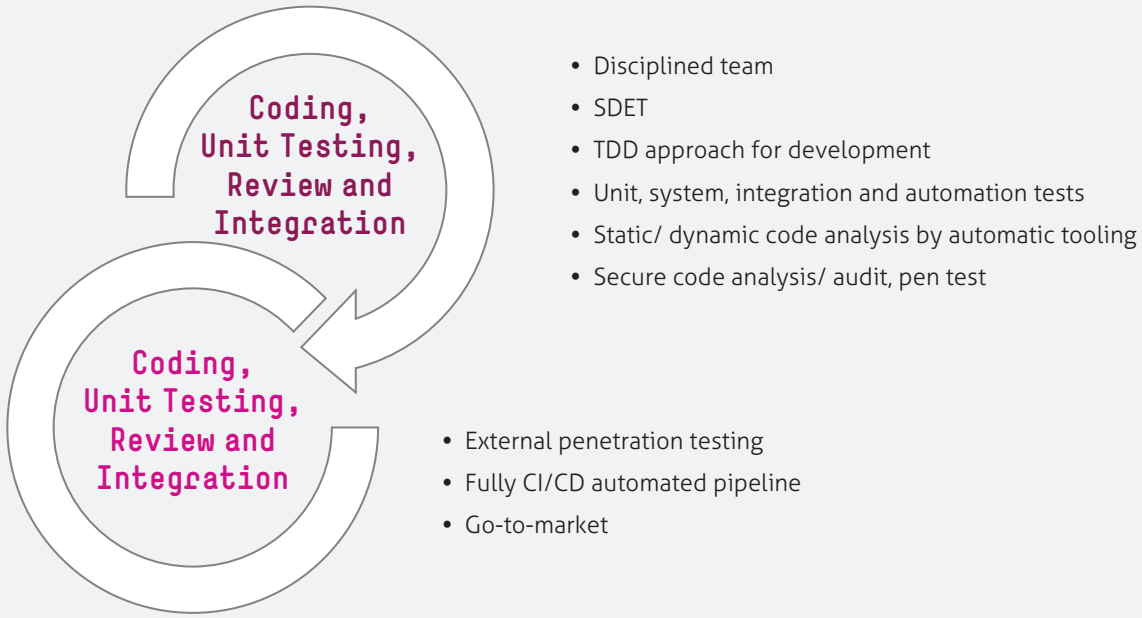


Figure 3 Modular Development Model

Relevance of development models

In this section, we have compared the pros and cons of various delivery models.

Comparison of various development models

While we have compared the assembly line, lean supervisor and modular development models, it is pertinent to note that all of them have their own pros and cons, depending on what you need.

	Assembly line model	Lean supervisor model	Modular development model
Pros	• Simple structure and quick to accomplish • Suitable for small-to-midsize projects • Easy to test and analyze the feature realization • Secure code analysis/ audit, pen test	• MVP is delivered quickly • Works great for large-scale products requiring constant updating and always delivers value based on a well documented product • Team is motivated to make every product feature perfect, and not just accomplish the tasks	• Works great for large-scale products requiring constant updating and always delivers value based on a well documented product • The resulting features are always better than the initial ones
Cons	• The project requirements should be well defined • Sequential nature of the project execution involves risk • Review and integration is a weak link • Heavy integration testing and automation in coverage is low	• The documentation needs to be precise and a skilled analyst is needed to ensure 100% understanding of the requirements • This approach is suited for highly-skilled developers • Tool costing is moderate to high	• Works great for large-scale products requiring constant updating and always delivers value based on a well documented product • The resulting features are always better than the initial ones

Use cases of development models

The following table describes the best suited use cases –

	Assembly line model	Lean supervisor model	Modular development model
Best suited	Small to midsize projects	Mid to large size projects	Large size projects
Development velocity	High	Moderate	Slow
Multi-vendor	Most used	Gaining adoption	Least used
Onboarding	Shortest	Moderate	Longest
Fluid requirements	Suited	Adoptable with automation	Least suited
Automation	Low	Moderate to high	Very high

Best Practices

The following are the tools that are used to enable a lean supervisor model:

- HP Fortify for secure code review
- SonarQube, ReSharper, Lint for Static code analysis
- Profiling, Memcheck, ValGrind for memory leaks
- Automation (Unit and UI testing)
- Obfuscation tools like Dot Obfuscator, DexGaurd, Themida

- Penetration testing tools like Burp Suite
- Confluence and Jira for project management
- Crucible, Jenkins for DevOps
- GitLab for source control management

Use cases of development models

Given below are the key best practices:

- The design should be properly documented.
- The team should identify all the dependencies and get the proper resolution plan from dependent team during planning.
- Product documentation should be driven from confluence wiki and requirements execution must be driven from JIRA.
- Each JIRA story should be linked to reviews and test cases. Once the code is successfully reviewed and all the test cases are successfully completed, we can close the JIRA story.
- JIRA XRAY plugin can be used for integrating the test cases directly to user stories to automatically generate the automated report.

We have depicted the key best practices for lean supervisory model in Figure 4

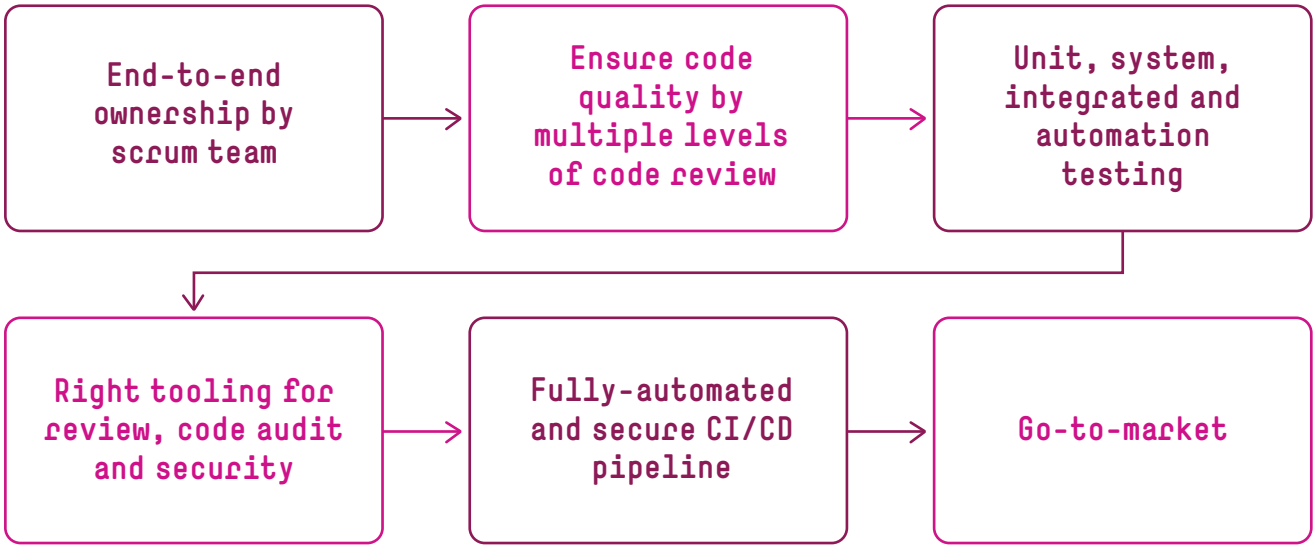


Figure 4 – Lean supervisory model best practices

References

1. https://medium.com/globalluxsoft/5-popular-software-development-models-with-their-pros-and-cons-12a486b569dc

About the authors

Ramesh Kumar

He is a mobile architect with around 14 years of IT experience. He has worked on multiple mobile application and SDKs, and has vast experience in design and developing an application from scratch in different technology variance apps like native, cross and hybrid applications. Ramesh has also streamlined teams towards effectively using secure and best dev practices and improved the fully-automated CI/CD pipeline, translating to correct, on time delivery.

Dr. Shailesh Kumar Shivakumar

He has 19+ years of experience in a wide spectrum of digital technologies including, enterprise portals, content management systems, lean portals and microservices. Dr. Shailesh holds a PhD degree in computer science and has authored eight technical books published by the world's top academic publishers such as Elsevier Science, Taylor and Franscis, Wiley/IEEE Press and Apress. Dr. Shailesh has authored more than 14 technical white papers, five blogs, twelve textbook chapters for various under-graduate and post graduate programs and has contributed multiple articles. He has published 20+ research papers in reputed international journals. Dr. Shailesh holds two granted US patents, apart from ten patent applications. Dr. Shailesh has presented multiple research papers in international conferences. Dr. Shailesh's Google Knowledge Graph can be accessed at https://g.co/kgs/4YoaiN . He has successfully led several large scale digital engagements for Fortune 500 clients. Shailesh can be reached at Shaileshkumar.Shivakumarasetty@mindtree.com

About Mindtree

Mindtree [NSE: MINDTREE] is a global technology consulting and services company, helping enterprises marry scale with agility to achieve competitive advantage. "Born digital," in 1999 and now a Larsen & Toubro Group Company, Mindtree applies its deep domain knowledge to 275+ enterprise client engagements to break down silos, make sense of digital complexity and bring new initiatives to market faster. We enable IT to move at the speed of business, leveraging emerging technologies and the efficiencies of Continuous Delivery to spur business innovation. Operating in more than 15 countries across the world, we're consistently regarded as one of the best places to work, embodied every day by our winning culture made up of over 22,000 entrepreneurial, collaborative and dedicated "Mindtree Minds."