

Devops

World without Devops

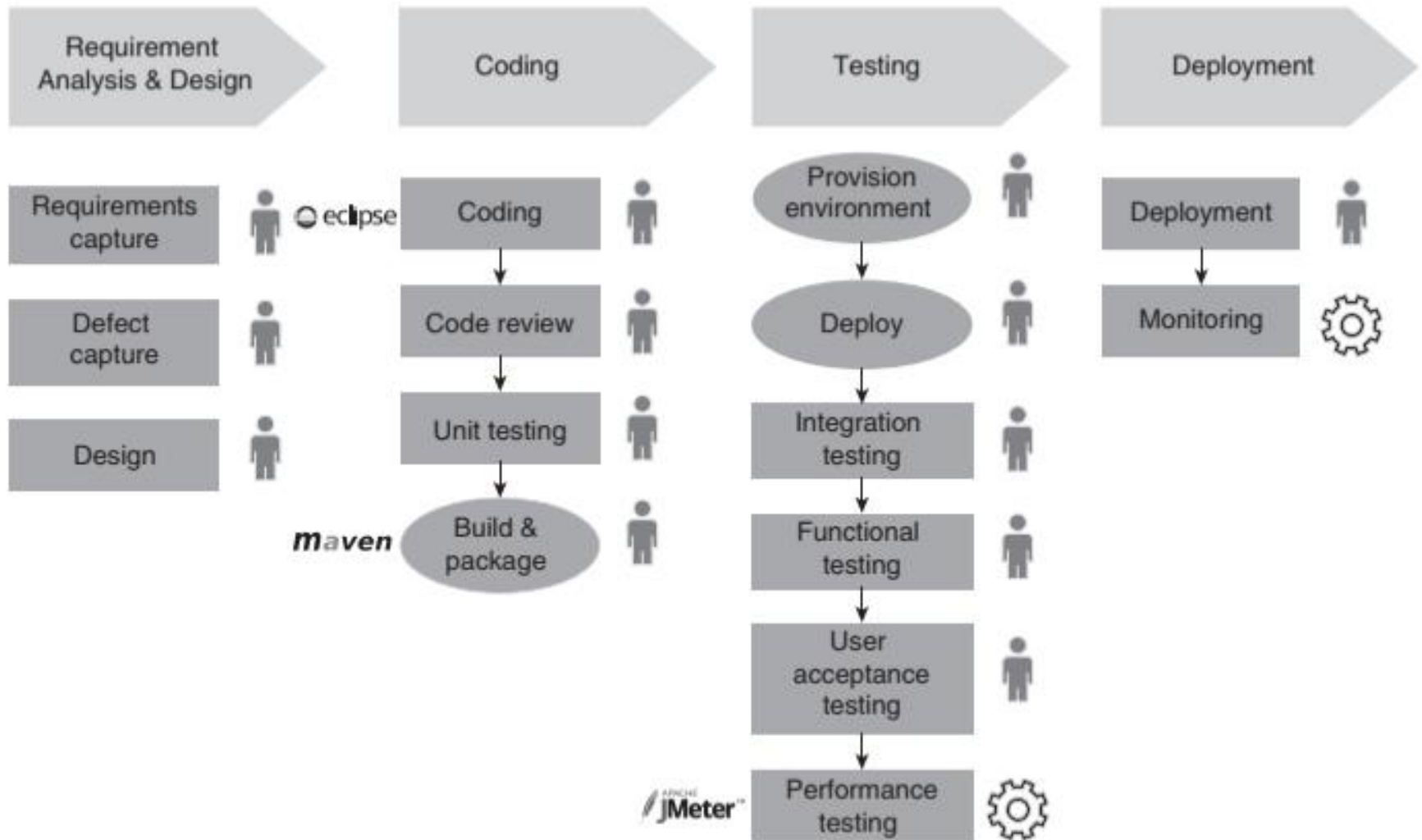
- DevOps
- what it is and how it works
- How are application development and maintenance carried out in this world?
- What is the team composition?
- What are the typical challenges that different stakeholders in these application teams face?
- What are the key concerns that affect applications
- How are delivery teams currently trying to find answers to these questions?
- How much success are they bagging currently?
- Which are the concerns still not having concrete answers?

PROBLEM CASE DEFINITION

- online share trading application
- 1. Shares are commoditized share in the ownership of a company.
- 2. Shareholders can own shares by buying those directly from the company or from other shareholders.
- 3. Place where buying and selling or trading of shares is done is called exchange.
- 4. Not only shares, but products derived from shares are also traded at exchanges.
- 5. Exchanges provide facility to shareowners to perform this activity of buying-selling shares and derived products through brokers.
- 6. Many brokers offer different channels to shareowners to perform trading activity, such as call centers, desktop application, mobile application, etc.
- 7. Shareowners can perform trading activity using these facilities provided by the broker on the exchange during exchange business hours.
- 8. Online share trading application, which is our case for discussion, is one such channel provided by the broker or investment bank to its customers, who are or want to be shareowners.
- 9. This is an internet-based application that the traders (or investors) can use from anywhere using a desktop or any other device that supports it.

- **Highlights of this trading activity that influence application development and maintenance?**
- 1. Trading is done during business/working hours of exchange.
- 2. The number of people trading is continuously varying and can surge due to any external event.
- 3. There are frequent minor or major modifications to the way trading is done.
- 4. Trading activity needs to be very secure.

Share trading application maintenance activity view.



- **1. Requirement Analysis:** Business users provide functional and non-functional requirements to the development team in a certain format.
- **2. Design:** Designers outline the structure of the program and document it to a certain extent.
- **3. Coding:** Developers write code as per design, perform peer code review followed by unit testing.
- **4. Testing:** There are four types of testing cycles, in most of the cases, before software is ready for production deployment – *integration testing, functional testing, user acceptance testing and performance testing*. Multiple teams perform these testing cycles. The operational team ensures that the software is packaged and deployed on a test environment for testing
- **5. Deployment:** TradeLive application is packaged and deployed on production environments and monitored.

Devops

- Most common interpretation of DevOps is Dev + Ops, that is, Development + Operation.
- DevOps concept covers technology, process, culture, etc. involved in improving collaboration amongst Development and Operation teams.

key challenges

- 1. There are many steps **involving manual actions** leading to dependency on individuals in the team.
- 2. As highly manual **release/deployment process starts post market closure** on Fridays, most of the time team works late night and over the weekend to accomplish it.
- 3. **Quality** is a concern at most of the stages in the development process.
- 4. There are multiple environments, and it takes a **very long time to provision** those before a testing cycle begins after making changes to TradeLive. This requires advance planning and sometimes delays testing cycle due to delays in provisioning activity.
- 5. **Traceability of requirement** through the development lifecycle stages is a tedious job.
- 6. **Repeatability** of testing cycles is very costly.
- 7. As the team is **new to agile** methodology, there are concerns that it will further **impact the quality of deliverables**, and agile benefits may not be realized fully.
- 8. Team is **not able to increase release frequency** though there is such demand from business.
- 9. **Management does not have the insight of deliverable quality** due to unavailability of reports.

BENEFITS OF FIXING APPLICATION DEVELOPMENT CHALLENGES

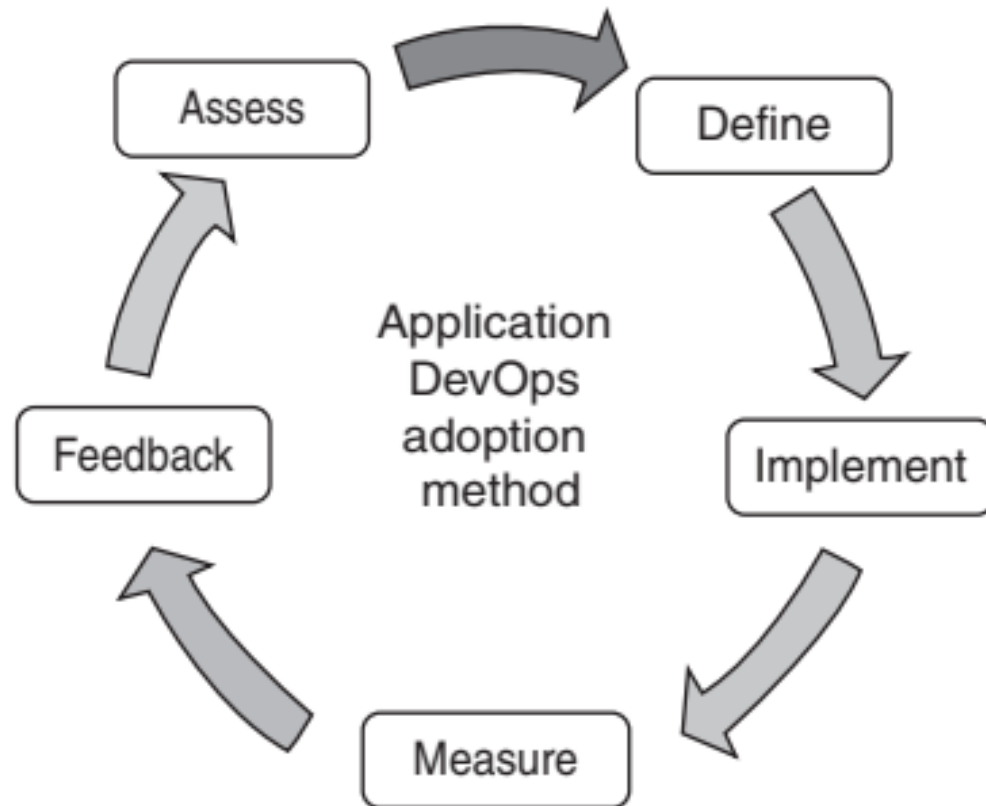
- **1. Time to Market:** How quickly can a change reach end users? When end users or business users demand certain functionality change in the application, how fast can IT turn it around and deliver it to production.
- **2. Implementation Cost:** Can we reduce the implementation cost? Cost has been a key player in software delivery.
- **3. Quality:** Does it improve and become predictable? There are two important parameters: actual delivered quality of software product and end quality predictability in early stages of development.

DEVOPS ADOPTION APPROACH THROUGH ASSESSMENT

- we know there are challenges, what do we do next?
- How can we find out all or most of the issues?
- The answer is - Experts can tell by reviewing application, processes, tools, etc.
- This process of reviewing the application and related processes to identify the status of DevOps and coming up with the recommendation is called assessment/maturity assessment.
- Five-step process to identify and fix application challenges.
 - 1. Assess to identify gaps.
 - 2. Define solutions.
 - 3. Implement solutions.
 - 4. Measure benefits.
 - 5. Feedback benefits and challenges.

- Improving application delivery process is not a one-time activity but rather an iterative process.
- implement the defined solutions, measure and feedback results
- next iteration implement the right solutions to the next set of problems

Application assessment process.



- 1. **Assess** to identify gaps in process, technology, tools and automation. Assessment step tries to identify all the issues that are affecting the software delivery.
- 2. **Define** solutions to the issues as per priority. Definition phase also comprises solutions as per organization's DevOps strategy.
- 3. **Implement** solutions in a step-by-step manner by taking maximum benefit, low impact recommendations first.
- 4. **Measure** benefits along with solution implementation so that the team can understand the success of the strategy.
- 5. **Feedback** benefits and challenges to refine solutions to achieve the right results in fewer iterations.

TIME TO MARKET

- Total time required to realize a requirement called as time to market.
- When business users request for a feature in the application, it goes through many lifecycle stages
- Ex; analysis, design, coding, testing, etc. before it is actually made available in the production environment for the end user. This time is called as time to market.
- Reducing this feature delivery time gives a competitive advantage to businesses.

SOLUTION DIMENSIONS

- challenges in the application will be fixed by introduction of a set of tools to automate development steps.
- must consider additional dimensions to realize actual business benefits because software development is a complex process involving multiple factors in addition to tools.
- three key dimensions
 - **1. Process:** It includes the development methodology, for example, waterfall to agile, operational processes, different approval processes to release a change to production.
 - In other words, any processes followed in realizing a requirement in the form of an application feature must be reviewed.

- **2. Tools:** Tools that can help to automate currently manual activities. Solution must focus on tools being used in the current delivery process, also propose a set of tools required in addition to the current toolset.
- **3. Architecture and Technology:** It is aligning architecture and technology to allow accelerated development.
- Solution must evaluate architecture from automation point of view. If the architecture is not capable of adapting to parallel development, automation and tools, then it needs refinement so that it does not become a hurdle in accelerated delivery.

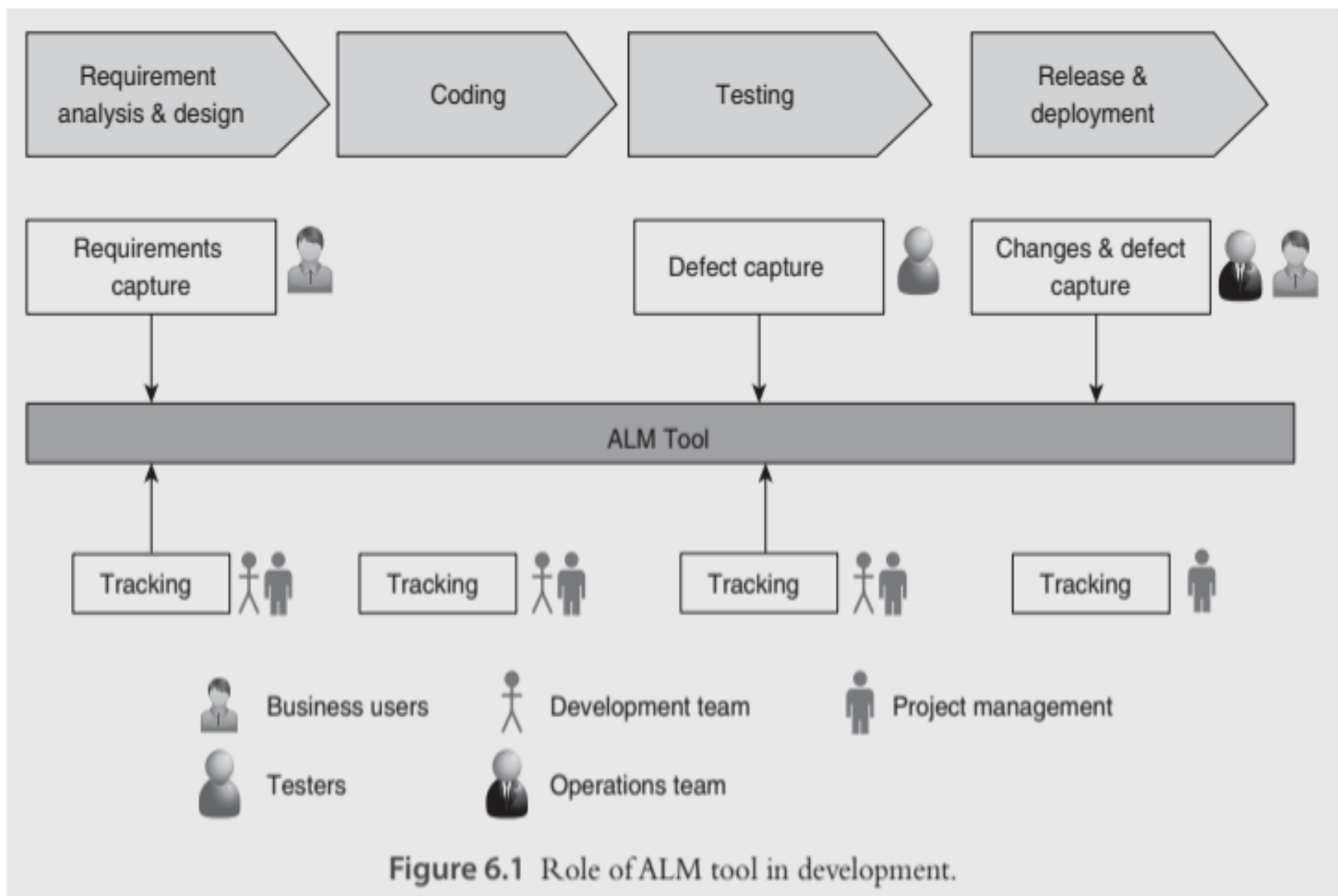
Application Lifecycle Management

- **Scenario: How do I automate tracking and collaboration amongst team members?**
- how do we track actual application lifecycle progress?
- When there are multiple developer teams working on different projects of same application, how do we ensure there is collaboration amongst these teams?
- Can a requirement be traced from requirement stage to production deployment?
- Can a requirement, during development lifecycle stages of it, be assigned to different responsible team?
- Many such questions can be answered by using an **Application Lifecycle Management (ALM)** tool.
- ALM tools give this capability to manage requirements, defects, test cases, etc.
- Each tool has a different level of maturity to support these activities.
- Few tools like JIRA allow integration with plugins to support additional features.
- Atlassian JIRA, which is a popular ALM tool now days.

Case: ALM for Trading Application

.

- List of key teams and their primary objective to use any ALM tool
- **1. Business Users:** Requirements (and defect) recording and tracking to closure.
- **2. Development Team:** Getting requirements, defects and updating progress of development to others.
- **3. Project Management:** Tracking project progress and tracing requirements.
- **4. Testers:** Getting requirements and reporting test progress and defects.
- **5. Operations:** Release progress and issue reporting. In



- It starts with requirements (say user stories) getting captured by business users.
- Developers and project management teams use this tool to update and track statuses whenever activities against requirement are completed.
- Teams can view progress by different teams on individual tasks of requirements in the ALM tool.
- Testing activity and monitoring results in feedback loop through ALM tool.
- Defects captured by testing team are tracked using ALM tool.
- Defects can be tracked against respective requirement
- Monitoring or post-production usage of application leads to product issues or changes in functionality. Both are tracked through ALM.
- There may be different workflows for fresh requirements, defects and production issues. ALM tools allow assignment of relevant workflow to the item type.

- ALM tools provide features so that these teams, with heterogeneous interests, can collaborate with each other on single ground.
- Teams can communicate amongst themselves using ALM tool continuously.
- **Workflow**
- a workflow is sequence of activities, statuses related to those activities, and changes in sequence of activities according to status of previous activity.

- List of functionalities expected from ALM tools
- 1. Requirement registering
- 2. Requirement traceability
- 3. Task tracking
- 4. Defect tracking in testing phases
- 5. Role-based access
- 6. Workflow
- 7. Reporting dashboards
- 8. Notifications
- 9. Support to different processes, for example, Waterfall, Agile, Kanban
- 10. Multi-location team support
- 11. Ability to integrate with other tools
- 12. Support to different types of project, for example. Development, Maintenance, Reengineering, etc.

JIRA

- JIRA is a proprietary Application Lifecycle Management (ALM) tool developed by Atlassian.
- It is mainly used for
 - 1. Project Management
 - 2. Requirements Management
 - 3. Process Management
 - 4. Release Management
 - 5. Customer Service
 - 6. Issue/Task Tracking
 - 7. Matrices Reporting

Key Features

- 1. It provides support for tool integration in CI/CD space so as to provide end-to-end traceability from backlog to deployment.
- 2. It is easily extensible with 1000+ third-party add-ons supporting JIRA extension so as to fit different use cases.
- 3. It provides customizable workflows so as to match individual release process.
- 4. It has rich APIs so as to integrate with other tools.
- 5. It has mobile apps so as to stay connected with the team from anywhere.

- Popular Competitors of JIRA
 - 1. CA Rally
 - 2. Microsoft Team Foundation Server
 - 3. HP ALM
- Where can JIRA be Best Utilized?
 - 1. General team project collaborations
 - 2. Change Requests Management
 - 3. Workflow Approvals
 - 4. Task Management
 - 5. On-boarding of employees
 - 6. Tracking of advertising campaigns

CHANGE REQUEST

- When a business user asks for change in requirement, it is done by raising a change request.
- Change request may undergo different lifecycle stages depending on size of change request.

Pros and Cons of JIRA

Pros	Cons
<ul style="list-style-type: none">• Supports security restricted issues.• Better searching using the distinct fields.• Good for tracking dependencies and relations by using link and sub-task feature.• Customization features as per project needs.	<ul style="list-style-type: none">• Complex initial setup and configurations.• Difficult to use, requires initial training time.• Version is associated to one JIRA project.

Example of Reference Architecture.

- JIRA is an extensible platform that can be **customized to match the business process**.
- It **improves productivity** by cutting down the time wasted on tracking issues and coordination.
- It fits well in the Continuous Planning phase of DevOps.
- There are different reference architectures possible with JIRA, as well as various CI/CD tools.

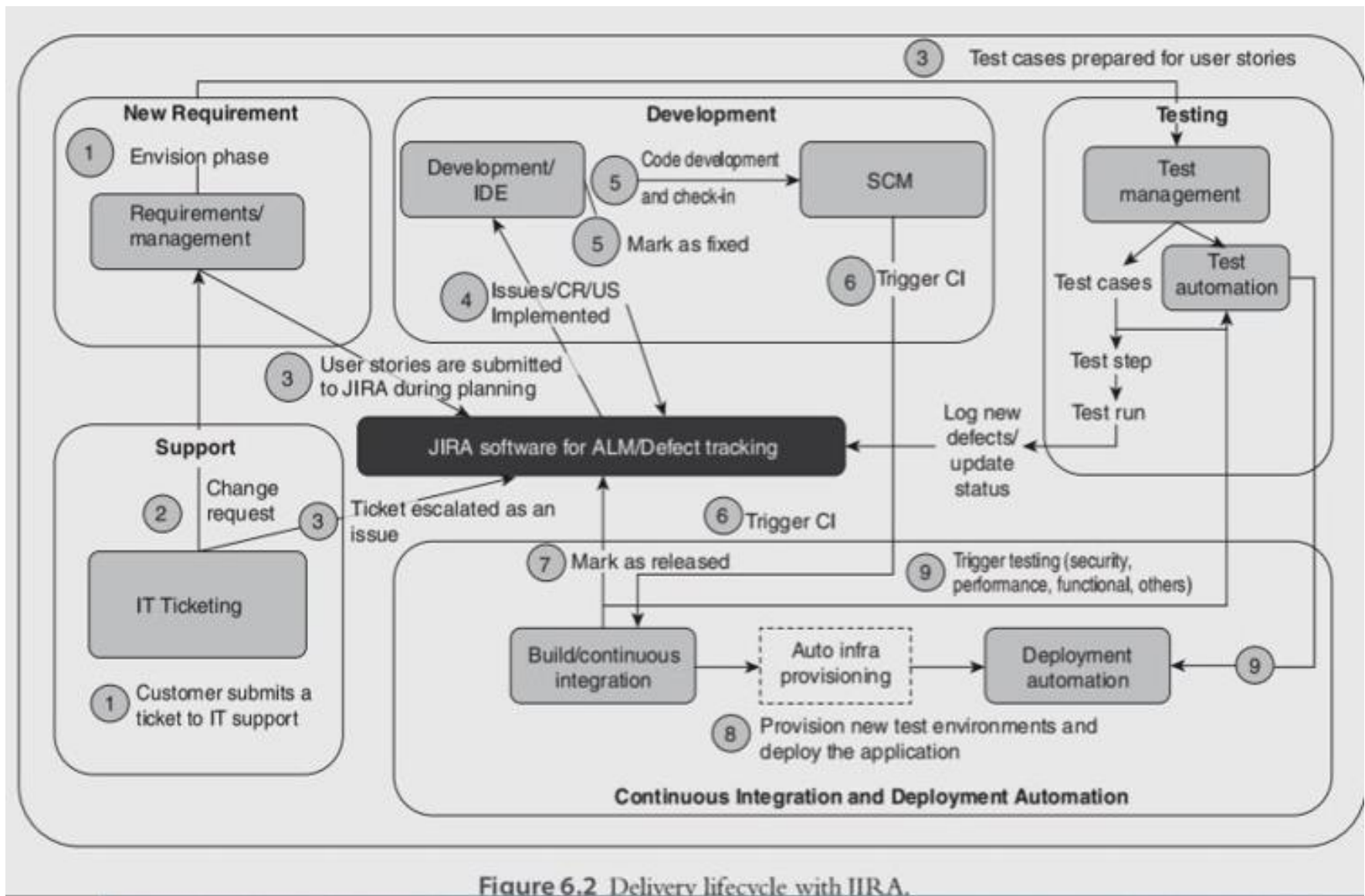


Figure 6.2 Delivery lifecycle with JIRA.

INTRODUCTION TO JIRA



INTRODUCTION TO JIRA



PLAN



COLLABORATE



EXECUTE



DOCUMENT



EVALUATE

WORKFLOWS IN JIRA



TO DO



IN PROGRESS



TESTING



DONE

WORKFLOWS IN JIRA



TO DO

TASK 1

TASK 2



IN PROGRESS



TESTING



DONE

WORKFLOWS IN JIRA



TO DO



IN PROGRESS



TESTING



DONE

TASK 1

TASK 2