# ♦ 18 Agile Development interview Q&A for Java developers

Posted on August 25, 2014 by Arulkumaran Kumaraswamipillai

Q1. What are the typical roles and responsibilities of an agile core team?

A1. The **Product Owner** represents the stakeholders and is the voice of the customer. He or she is accountable for ensuring that the team delivers value to the business. The Product owner writes typically the user stories, prioritizes them, and adds them to the product backlog.

The **Development Team** is responsible for delivering potentially shippable product in increments at the end of each Sprint. A Development Team is made up of 3–9 people with cross-functional skills who do the actual work (analyze, design, develop, test, technical communication, document, etc.). The development team consisting of developers,

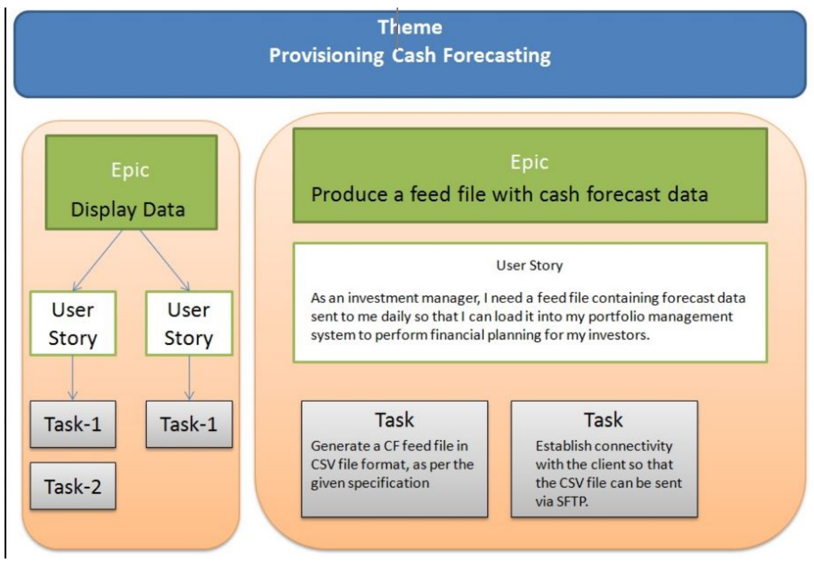
600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all

Ice Breaker Interview
Core Java Interview (
JEE Interview Q&A (3
Pressed for time? Jav
Job Interview Ice B
FAQ Core Java Jol
FAQ JEE Job Inter
FAQ Java Web Ser
Java Application Ar
Hibernate Job Inter
Spring Job Intervie
Java Key Area Ess
♦ Design pattern
♥ Top 10 causes
♥♦ 01: 30+ Writir
♦ 12 Java desigr
♦ 18 Agile Devel
♦ 5 Ways to debu
♦ 9 Java Transac
♦ Monitoring/Pro
02: ♥♦ 13 Tips tc

testers, operations staff, etc define the definition of "Done", which includes non-functional requirements like security, performance, cross browser compatibility testing, etc. Generally, you will have the doers and the helpers in the development team. The Development Team in a Scrum is self-organizing.
**Scrum Master** is a facilitator, and is accountable for removing impediments to the ability of the team to deliver the sprint goal/deliverables. The Scrum Master is not the team leader, but acts as a buffer between the team and any distracting influences. The Scrum Master ensures that the Scrum process is used as intended. The Scrum Master is the enforcer of rules. A product manager cannot be the scrum master.

Q2. What do you understand by the terms epic and user story?
A2. An agile **Epic** is a group of related **user stories**. For example, an online shopping cart app will have epics like user administration, product management, and shopping experience. An epic will have a number of broken down user stories. For example, the "user administration" epic can be broken down into user stories like "add, modify, and delete user", "manage user passwords", "generate raw data downloads", "generate aesthetically pleasing reports", etc.



As an investment manager, I need a feed file containing forecast data sent to me daily so that I can load it into my portfolio management system to perform financial planning for my investors.

Generate a CF feed file in CSV file format, as per the given specification

Establish connectivity with the client so that the CSV file can be sent via SFTP.

Agile theme, epic, story and task

15 Security key
4 FAQ Performa
4 JEE Design Pa
5 Java Concurre
6 Scaling your Ja
8 Java memory
OOP & FP Essenti
Code Quality Job I
SQL, XML, UML, JSC
Hadoop & BigData Int
Java Architecture Inte
Scala Interview Q&As
Spring, Hibernate, & I
Spring (18)
Spring boot (4)
Spring IO (1)
Spring JavaConf
01: ♥♦ 13 Spring
01b: ♦ 13 Spring
02: ► Spring DII
03: ♥♦ Spring DI
04 ♦ 17 Spring b
05: ♦ 9 Spring Be
06: ♥ Debugging
07: Debugging S
Spring loading p
Hibernate (13)
01: ♥♦ 15+ Hiber
01b: ♦ 15+ Hiber
02: Understandi
03: Identifying ar
04: Identifying ar
05: Debugging H
06: Hibernate Fi
07: Hibernate mi
08: Hibernate au
09: Hibernate en
10: Spring, Java
11: Hibernate de
12: Hibernate cu
AngularJS (2)

— A **theme** is an objective that may span projects and products. Themes can be broken down into sub-themes, which are more likely to be product-specific. At its most granular form, a theme can be an epic.
An epic is a group of related user stories. An epic needs to be broken down into a user story before it is sized using fibonacci number (eXtra Smal [XS=1 pt] Small [S = 2 pt], Medium [M = 3], Large (L = 5), XtraLarge (XL = 8) , XtraXtraLarge [XXL = 13pts], XtraXtraXtraLarge [XXXL = 21 pts] ) and introduced into a sprint. The sizing is similar to the T-shirt sizes. These points are also known as the velocity points to monitor progress.
— A **user story** is an Independent, Negotiable, Valuable, Estimatable, Small, Testable requirement, which is abbreviated to "INVEST". Even though stories are independent, they have no direct dependencies with other requirements and user stories may be combined into epics when represented on a product road map. User stories need to be defined prior to sprint planning in ABC (Accelerated Business Case) sessions where the development teams, QA (i.e. testing) teams and product managers can use them to discuss, size and prioritise at sprint-level. User stories will often be broken down into tasks during the sprint planning Process unless the stories are small enough to be consumed on their own. Tools like "MindMap" is used to capture the stories in the form of **COS** (i.e. Condition Of Staisfaction).
— The sized cards are prioritized interms of its value add to achieve the MVP (Minimum Viable Product) defined by the product manager and the business owners.

The user stories are in the format of

"As **Who** I want **What** so that **Why**"

Here is an example.

"As an Investment Manager I want to be able to view client account balances, so that I can make informed investment decisions. "

⊞ Git & SVN (6)
⊞ JMeter (2)
⊞ JSF (2)
⊞ Maven (3)
⊞ Testing & Profiling/Sa
⊞ Other Interview Q&A f
⊞ ▶Free Java Interviev

## 16 Technical Key Areas

open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3
⊞ Java Debugging (21)
⊞ Judging Experience I
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Managen

## 80+ step by step Java Tutorials

open all | close all
⊞ Setting up Tutorial (6)
⊞ Tutorial - Diagnosis (2
⊞ Akka Tutorial (9)
⊞ Core Java Tutorials (2
⊞ Hadoop & Spark Tuto

"As a support staff I want the system to allow me run multiple searches at the same time, so that I can do my job faster. "

Q3. What do you understand by the term "Condition Of Satisfaction (aka COS)"?
A3. For each story, the business user, analyst, owner, or stake holder will define the "**condition of satisfaction (COS)**" criteria to satisfy that particular story. You can also think of it as an "acceptance criteria". A COS questions the User Story, and encourages conversation between the Product Owner and the team. A good way of gathering COS is asking questions such as:

— 'What if … ?',
— 'Where …?',
— 'When …?',
— 'How …?'.

The test cases will be written by the testers and development tasks will be carried out by the developers to satisfy the COS. To be accepted, the development task should also satisfy other non functional tasks like writing unit tests, continuous integration and build, security, performance, data archival, etc. The story will also have to be fully tested. For example, COS will look like

– A search functionality to be able to search clients by client code.
– Display the client list sorted in alphabetical order.
– Ability to click on a particular client to display his or her account balances.
– The account balances need to be sorted by account type.
….and so on including non functional requirements as well.

Q4. How do you estimate development and testing effort of a user story?
A4. Agile projects normally have a number of **sprints** to finish the user stories. Each sprint will be of 2-3 weeks. One or more user stories are allocated to each sprint. The developers and testers will estimate on it to have story developed and fully tested to satisfy the COS. If a story is big

⊞ JEE Tutorials (19)
⊞ Scala Tutorials (1)
⊞ Spring & HIbernate Tu
⊞ Tools Tutorials (19)
⊞ Other Tutorials (45)

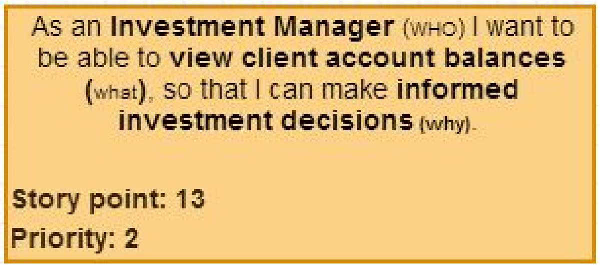
## 100+ Java pre-interview coding tests

open all | close all
⊞ Can you write code? (
⊞ ♦ Complete the given
⊞ Converting from A to I
⊞ Designing your classe
⊞ Java Data Structures
⊞ Passing the unit tests
⊞ What is wrong with th
⊞ Writing Code Home A
⊞ Written Test Core Jav
⊞ Written Test JEE (1)

## How good are your …..?

open all | close all
⊞ Career Making Know-
⊞ Job Hunting & Resum

enough so that cannot be completed within a sprint (i.e. in 2 weeks), that particular sub story can be split further into 2 or more stories.

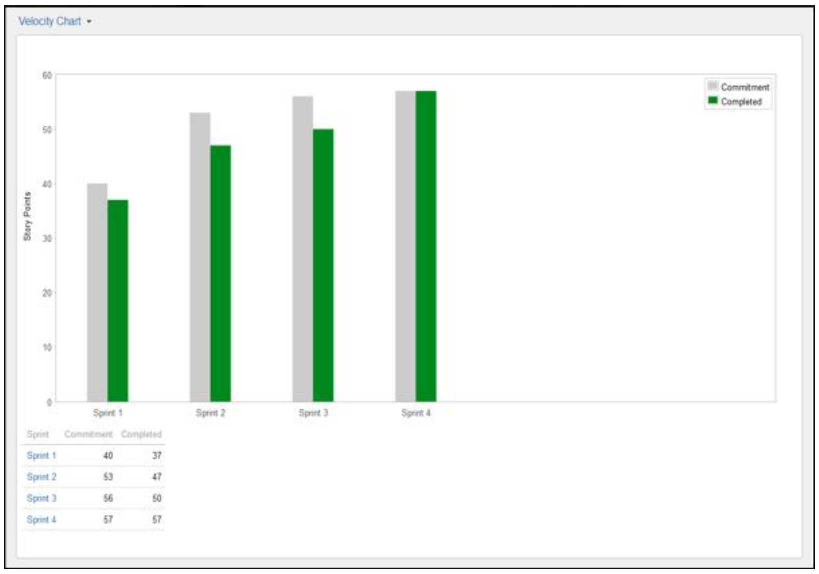


As an **Investment Manager** (WHO) I want to be able to **view client account balances** (what), so that I can make **informed investment decisions** (why).

**Story point: 13**
**Priority: 2**

agile story with points allocated

The stories are estimated by allocating points. It is known as the “velocity points”. The velocity points are allocated to each story. Stories are tagged like T-shirt sizes — Small, Medium, Large, and Extra Large, etc. Each of these sizes are allocated velocity points using the Fibonacci series values. For example, 3, 5, 8 and 13. Where Small is given 3 points, Medium is given 5 points, Large is given 8 points, and extra large is given 13 points. A the end of each sprint, the velocity points for all the user stories are added and reported to the management in a “showcase” to monitor progress. The points play(i.e. the user stories that are not completed within this sprint) that are added to the next sprint.



| Sprint | Commitment | Completed |
|---|---|---|
| Sprint 1 | 40 | 37 |
| Sprint 2 | 53 | 47 |
| Sprint 3 | 56 | 50 |
| Sprint 4 | 57 | 57 |

agile velocity diagram

In the diagram above, you can see "commitment" and the "completed".

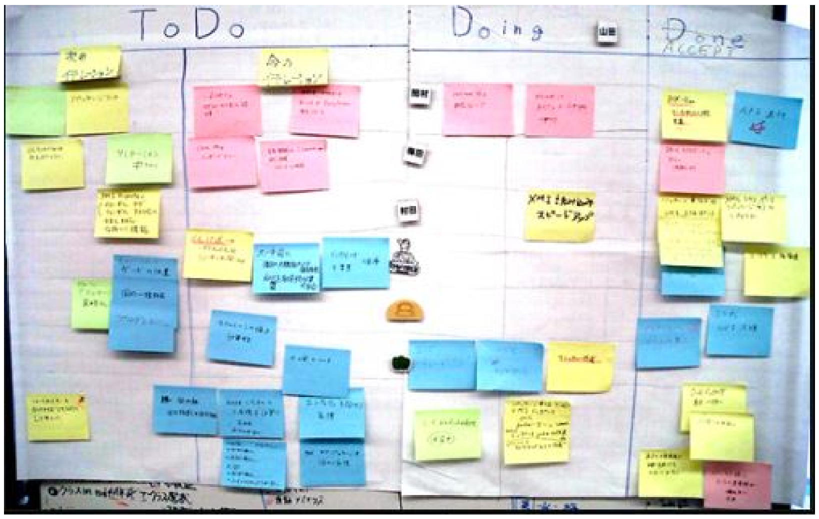
Q5. What are some of the key features and objectives of the daily stand-ups?

A5. Daily Stand-ups are named literally after the way the meeting is conduced — i.e. while standing up. This method helps ensure the meetings remain time-boxed to 15 minutes and not dragged on.

**The general features are:**
– It is a ritual that happens at the same time, in the same location, every day.
– The meeting is led by the Scrum Master, who keeps all attendees focused on answering the following key – questions to meet the project objectives.

1) What have I achieved since the last Stand-up?
2) What do I intend to do before the next Stand-up?
3) What are my impediments to making progress?

Here is the picture of user stories stuck to a board, and this is where the daily "stand ups" or "scrum sessions" take place.



agile stand up board

Q6. What are some of the key features of agile development?

A6.

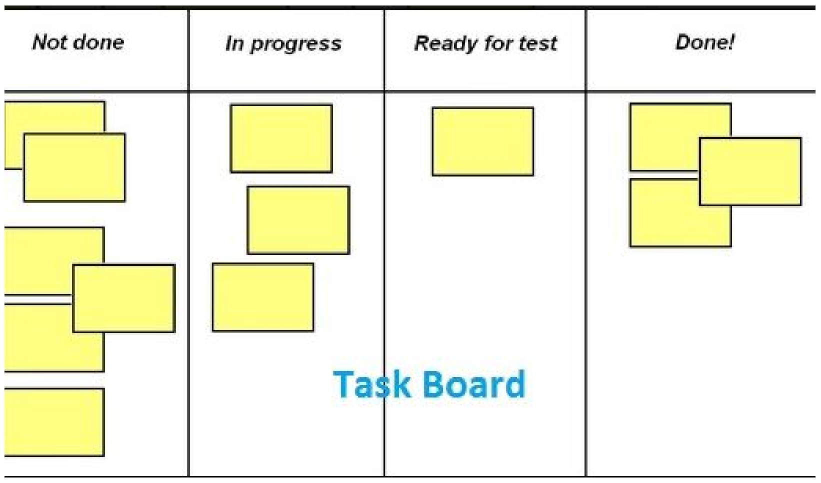
– Collective code ownership and freedom to change with proper unit tests, integration tests, and continuous build & integration.
– Incremental approach (e.g. user stories are incrementally implemented)
– Automation (e.g. TDD — Test Driven Development and continuous build & integration).
– Customer focused (for e.g. internal and external users and business analysts are your immediate customers)
– Design must be simple. Designing is an ongoing activity with constant re-factoring to achieve the rules of code simplicity like no duplication, verified by automated tests, separation of responsibilities, and minimum number of classes, methods, and lines.

Q7. How do you know that you are using agile development?
A7. You are using an agile practice when

– You have daily stand-up meetings.
– You use CRC (Class Responsibilities and Collaborators) cards.
– You use timeboxed task boards.
– You use TDD (Test Driven Development) or BDD (Behavior Driven Development), Continuous Integration, regular code reviews, pair programming, automated builds, continuous deployment and delivery, etc.
– You have iteration planning meetings and carry out iterative development.

Q8. What is a task borad?
A8. It is generally a white board divided into 3 sections — To Do, In Progress, and Done. Each task is written on a sticky note, and moved from one section to another to reflect the current status of the tasks. The task board is frequently updated, especially during the daily stand up meetings. Different layouts can be used.
Each task allocated to each team member is timeboxed. You can have variation to the layout as shown below and each sticky not can have points that add up towards the velocity

points (calculated by adding up the estimates of the features, user stories, requirements or backlog items that are successfully delivered in an iteration. ).



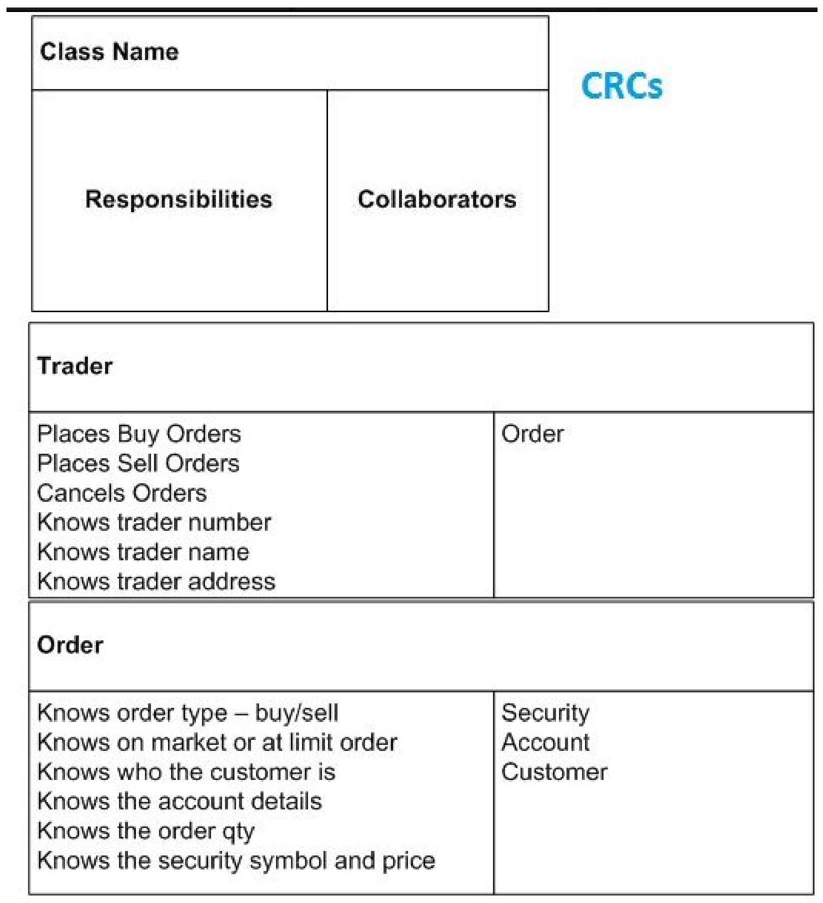
Agile task board

The task board is also known as the kanban board. Kanban is a Japanese word meaning card or sign. Each card or sign is equated with a user story. Whenever a particular user story is blocked for whatever reason, then the priority is to clear current work-in-process with the help of other team members to help those working on the activity that's blocking the flow.

Q9. What do you understand by the agile term timeboxed?
A9. A timebox is a previously agreed period for a particular task to be completed by an individual or team. The key aspect of the timebox approach is that stopping of work when the time limit is reached and evaluating what was accomplished instead of allowing the work to continue until the goal is reached, and evaluating the time taken.

Q10. What are CRC cards?
A10. CRC stands for Class, Responsibilities, and Collaborators. It is used for rapidly sketching an Object Oriented design and playing out the roles and responsibilities to validate the design. The role play dialog will be something like

Hello, I am a trader and responsible for placing and cancelling buy and sell orders on behalf of my customers. Before placing a trade, I must know my trader details like number, name, address. I need to collaborate with order to fill in the relevant order details.



CRCs

| Class Name | |
|---|---|
| Responsibilities | Collaborators |

| Trader | |
|---|---|
| Places Buy Orders<br>Places Sell Orders<br>Cancels Orders<br>Knows trader number<br>Knows trader name<br>Knows trader address | Order |

| Order | |
|---|---|
| Knows order type – buy/sell<br>Knows on market or at limit order<br>Knows who the customer is<br>Knows the account details<br>Knows the order qty<br>Knows the security symbol and price | Security<br>Account<br>Customer |

Agile – CRC

Q11. What do you understand by the term “collective ownership”?

A11. Collective ownership, as the name suggests, every team member is not only allowed to change other team member’s code, but in fact has a responsibility to make changes to any code artifact as necessary. This means every developer will review code written by others when integrating others’ changes from the code repository into their code to familiarize themselves and to identify any potential issues and mistakes. Every developer will be motivated to check in the code progressively and incrementally with proper automated unit and integration test cases as part of the continuous code integration.

Q12. What do you understand by the term Behavior Driven Development (BDD)?
A12. Behaviour-Driven Development (BDD) is an evolution in the thinking behind Test Driven Development (TDD — Writing tests before writing code) and Acceptance Test Driven Development (ATDD — write acceptnce tests, and for many agile teams, acceptance tests are the main form of functional specification and the formal expression of the business requirements). The BDD basically combines TDD and Domain Driven Design. It aims to provide common vocabulary that can be used between business and technology.

The acceptance tests are generally written using the "Given-When-Then" approach. For a given story/context, when some action is carried out, then a set of observable consequences should be obtained. For example, Given that you have enough available cash, when you place a trade within your available cash, then placing of your trades should succeed without any errors.

Q13. What do you understand by the terms user stories, story mapping and story splitting?
A13. User stories: Dividing up of the customer's or product owner's requirements into "functional increments" so that it can be worked on via the task board. This is done in consultation with the customers, product owners, or business analysts.

Story mapping: When you have a backlog full of user stories, you can select a few of them to work on during the next iteration. This step involves ordering of the user stories. The "map" arranges user tasks along the horizontal axis in rough order of priority and the vertical axis addresses the implementation details. It can be done either on a white board with sticky notes or using tools like Silver Stories.

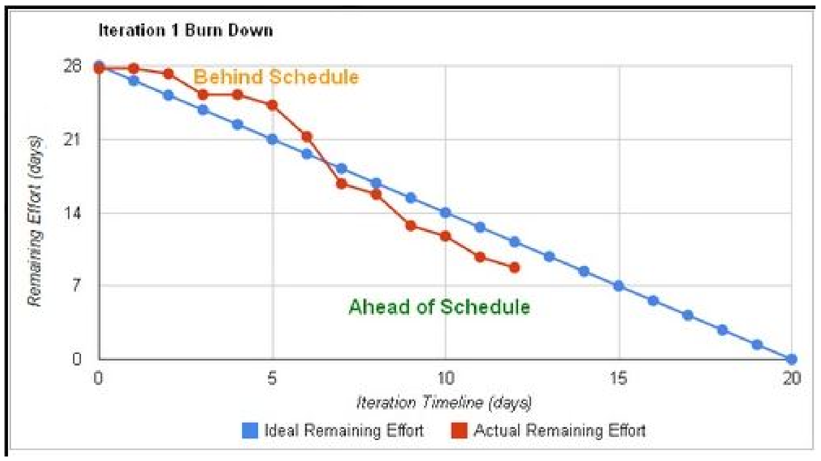
Story splitting: Before a story is ready to be scheduled for implementation, it needs to be small enough to pass the usual rule of thumb that "a story should be completed within the iteration". So, "story splitting" consists of breaking up one user story into smaller ones, while preserving the property that each user story separately has measurable business value.

Q14. How is an agile project monitored for progress?
A14. Any one who has worked on an agile project would have come across thes key terms:

Colocation
Frequent delivery
Daily standups
Timeboxed tasks

The best evidence that a software project is on track is working software, preferably deployed to production.

Secondly, a burn down chart is used to present the progress to the management. Each sprint is basically 2 weeks and this is plotted on the X axis. The Y axis will have the velocity points. In each Sprint depending on the team size, a number of sized up cards are picked and the individual points are added up. For example say, 40 points. This chart is about how quickly you burn through the stories. These graphs can be plotted manually or via tools like Excel spreadsheet or JIRA.



agile velocity graph or burn down chart

Velocity is the key to agile project management.

Q15. What are different roles in an agile project?
A15.

**Mandatory roles:**

**Team lead:** This role, called "Scrum Master" in Scrum or team coach or project lead in other methods
**Team member:** This role, sometimes referred to as developer or programmer, is responsible for the creation and delivery of a system. This includes modeling, programming, testing, and release activities, as well as others.
**Product owner:** The product owner is responsible for the prioritized work item list
**Stakeholder:** is anyone who is a direct user, indirect user, manager of users, senior manager, and operations staff member.

**Optional roles that are typically adopted only on very complex projects**

**Technical experts:** Sometimes the team needs the help of technical experts, such as build masters to set up their build scripts or an agile DBA to help design and test their database.
**Domain experts:** Sometimes the product owner will sometimes bring in domain experts to work with the team
**Independent tester:** to validate functional and non-functional testing. For example, security testing, performance testing, cross browser compatibility testing, etc.

Q16. What are the key benefits of agile mehodology?
A16.

– Decreased time to market as agile process is based on the philosophy of early and regular releases. The iterative nature of agile development means features are delivered

incrementally, enabling some benefits to be realised early as the product continues to develop.
– Better quality as testing is integrated throughout the sprints, 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.
– Improved communications due to active involvement and colocated multi-disciplinary teams. This provides excellent visibility for key stakeholders, both of the project's progress and of the product itself, which in turn helps to ensure that expectations are effectively managed.
– Lower costs because unlike in traditional development projects, where you write a big spec up-front and then tell business owners how expensive it is to change anything, particularly as the project goes on. In fear of scope creep and a never-ending project, we resist changes and put people through a change control committee to keep them to the bare minimum. Agile development principles are different. In agile development, change is accepted. Instead of rejecting changes, the timescale is fixed and requirements progressively emerge and evolve as the product is developed.

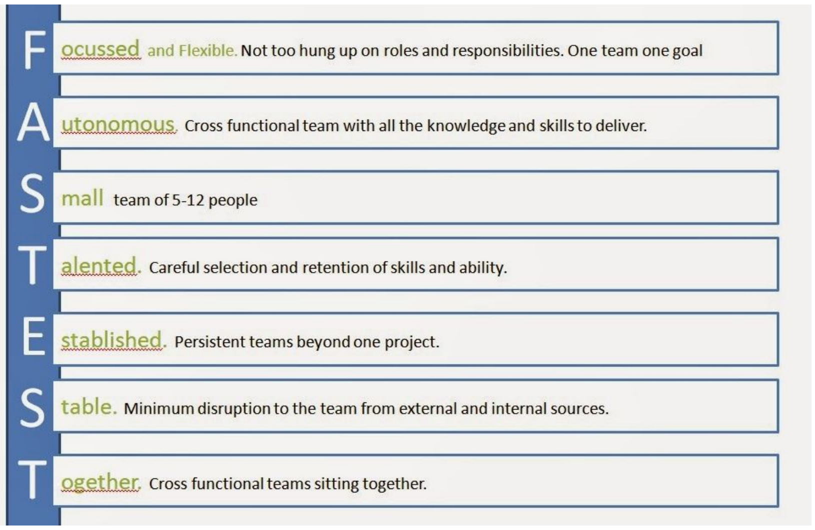
Q17. It looks too good, now what are the drawbacks of agile?
A17.

– Agile is a very sophisticated process and requires full management commitment and requires experienced and dedicated teams to achieve things within the allocated sprints. Otherwise, everything will blow up to become a backlogs.
– Time boxing can encourgage agile members to cut corners in terms of unit test coverage, documentation, code quality, design , failing to test negative scenarios, etc. It can also lead to over worked staff who lose motivation.
– It assumes that all developers are created equal. Doing a particular task might be a 5 point task for developer A, whereas for developer B, it takes 20 points. For example, this can happen if developer B is a back-end developer and not experienced with GUI development.

– Not all projects nor all phases are good candidate for agile. You should use agile if you can deliver tangible releases that can be tested by the testers within the sprint. For example, a GUI screen. Some projects may spend months developing back-end architecture without any tangible deliveries. Website development is a good candidate for agile development. It is also not suited where different teams use different methodologies. What happens when a development and QA team adheres to the principles of Agile, but the platform and production support do not?

So, there are pros and cons and some organizations make use of the hybrid approach to get the best of both worlds. Agile is commonly believed to be a set a practices, processes and tools, when in fact, Agile is really more of a mind-set and culture.

Q18. What are the agile resource Management principles?
A18. The resourcing principles can be abbreviated to FASTEST as shown below.



F ocussed and Flexible. Not too hung up on roles and responsibilities. One team one goal

A utonomous. Cross functional team with all the knowledge and skills to deliver.

S mall team of 5-12 people

T alented. Careful selection and retention of skills and ability.

E stablished. Persistent teams beyond one project.

S table. Minimum disruption to the team from external and internal sources.

T ogether. Cross functional teams sitting together.

Agile resourcing principles

Agile projects require require experienced developers and testers to get things done in allocated sprints. More and more organizations are getting into agile methodology, and it is really worth learning and experiencing it. I was fortunate enough to work on a number of such projects. These projects do need full commitment and support of the senior management.

## Popular Posts

♦ 11 Spring boot interview questions & answers

**827 views**

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

**767 views**

18 Java scenarios based interview Questions and Answers

**400 views**

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

**389 views**

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

**296 views**

♦ 7 Java debugging interview questions & answers

**293 views**

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

**286 views**

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

**279 views**

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

**240 views**

001B: ♦ Java architecture & design concepts interview questions & answers

**202 views**

Bio | **Latest Posts**

### Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

**About** Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ 04: JSP overview interview questions and answers

03: ♥♦ Spring DIP, DI & IoC in detail interview Q&As ›

**Posted in** Java Key Area Essentials, member-paid, SDLC

## Empowers you to open more doors, and fast-track

### Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

### Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

## Prepare to succeed

☀ Turn readers of your Java CV go from "Blah blah" to "Wow"? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

## © Disclaimer