

Agile methodologies



Kiran Juthi

Working at Mentor

Hyderabad, Telangana, India · [Contact info](#)



Mentor Graphics

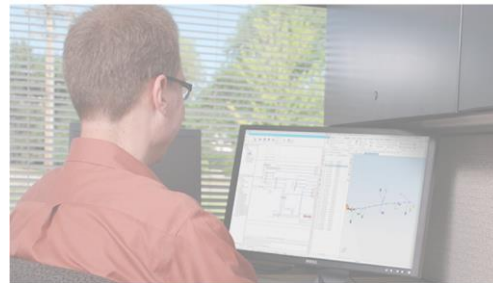


Indian Institute of
Technology, Madras



Capital

A comprehensive E/E systems development solution to efficiently engineer today's smart products. It covers the design, manufacture, and service of electrical systems as well as encompassing E/E system and software architectures, network communications, and embedded software development.



VeSys

VeSys is a suite of wiring and harness design software tools to satisfy the demanding requirements of companies where ease-of-use and value are as important as functionality.

<https://www.plm.automation.siemens.com/global/en/products/>

Academics vs industry

Exception handling

Domain object

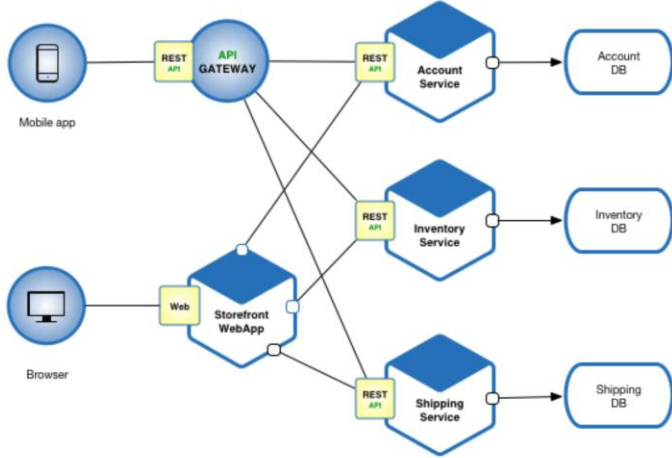
3rd party library

Configuration/
moving parts

DB variants

```

23 public class JdbcSample
24 {
25     public static void main(String[] args) throws ClassNotFoundException, SQLException
26     {
27         ElectricalObject electricalObject = new ElectricalObject()
28             .setType("Splice")
29             .setPinCount(2);
30
31         Class.forName("oracle.jdbc.driver.OracleDriver");
32         try (Connection connection = DriverManager.getConnection(
33             "jdbc:oracle:thin:@localhost:1521:ORCL", user: "SCOTT", password: "TIGER"
34         )) {
35             try (PreparedStatement statement = connection.prepareStatement(
36                 sql: "insert into electrical_obj value (?, ?)"
37             )) {
38                 statement.setString( parameterIndex: 1, electricalObject.getType());
39                 statement.setInt( parameterIndex: 2, electricalObject.getPinCount());
40                 statement.executeUpdate();
41             }
42         }
43     }
44 }
    
```



<https://microservices.io/patterns/microservices.html>

Encapsulation
Error reporting
Licensing

Abstraction
Quality
Performance

Scalability
Deployment

Security
Internationalization

Industry code base

Our code base is **5M+** lines of code in #java. New tools getting added in JavaScript/TypeScript. Mono repo is now **70GB** and growing. Clean build takes 50 minutes. Providing a faster feedback to **300+ engineering** team is our raison d'être.

Unit tests runs in ~72 hours. Our infrastructure runs them in parallel to get result in 2 hours. **100+ jobs** runs every day.

Our GUI Test framework based on Jimmy. **20K tests** requires 2000 hours-- Runs parallel using Univa Grid Engine to get overnight result.

Installers created, deployed and tested on hundreds of machines.

The infrastructure includes hundreds of servers and thousands of VMs. Maintained using Java, Ansible, Terraform, Jenkins(pipelines) etc.

Other tools of the trade: Perl, Python, Vue.js, Kafka, PostgreSQL, InstallAnywhere, Elastic, Grafana, Springboot, ant, Linux , Prometheus, vRealize...

Recent/Current projects: Migration to Java11, Kafka based data collection, Migration to Gradle to make code base more modular and maintainable, Micro service based infrastructure monitoring and healing, static code analysis to improve efficiency of impacted tests, Web based GUI for self-service, run entire test on demand on cloud like environment(public of private)....

Hypothetical e-commerce application

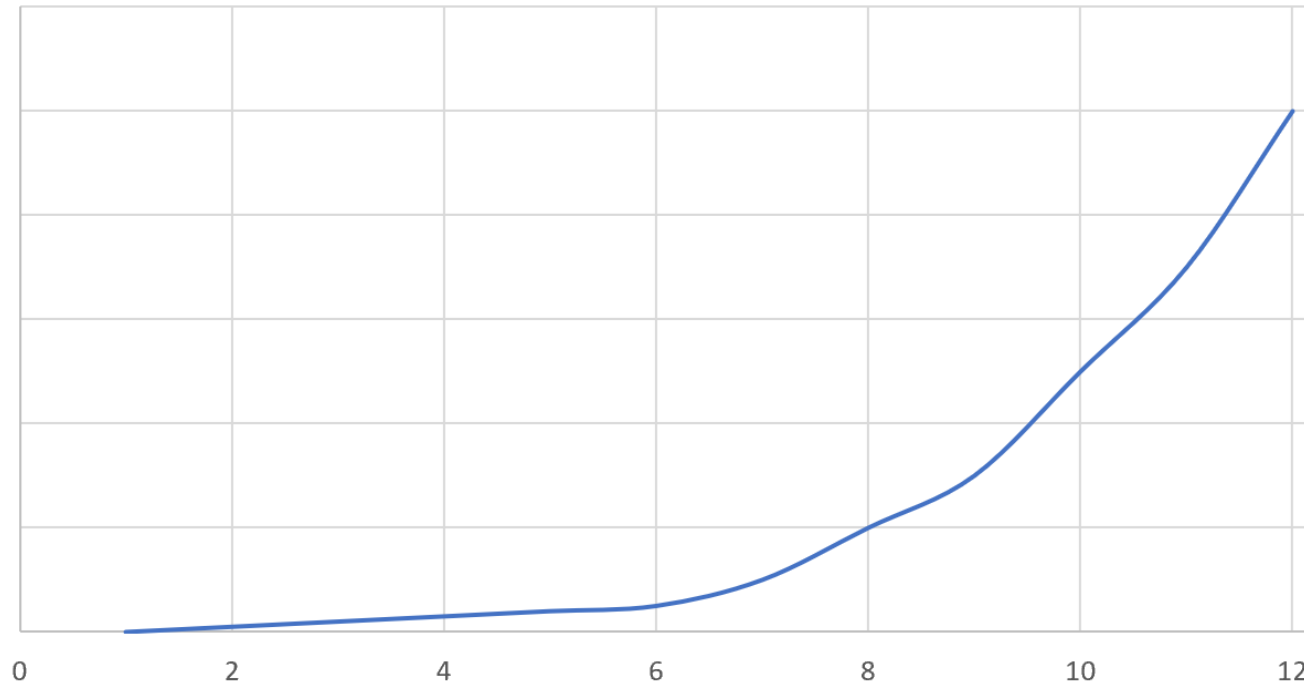
- 1 year release cycle
- In next release
 - User should be able to search for products
 - Search categories
 - Search filters
 - Payment support
 - Net banking
 - Debit and credit cards

Hypothetical e-commerce application...contd.

- Search functionality
 - Via QR code input
- Payment support
 - Payments via UPI
- New requirement
 - Contactless delivery via OTP

Hypothetical e-commerce application...contd.

Stress level



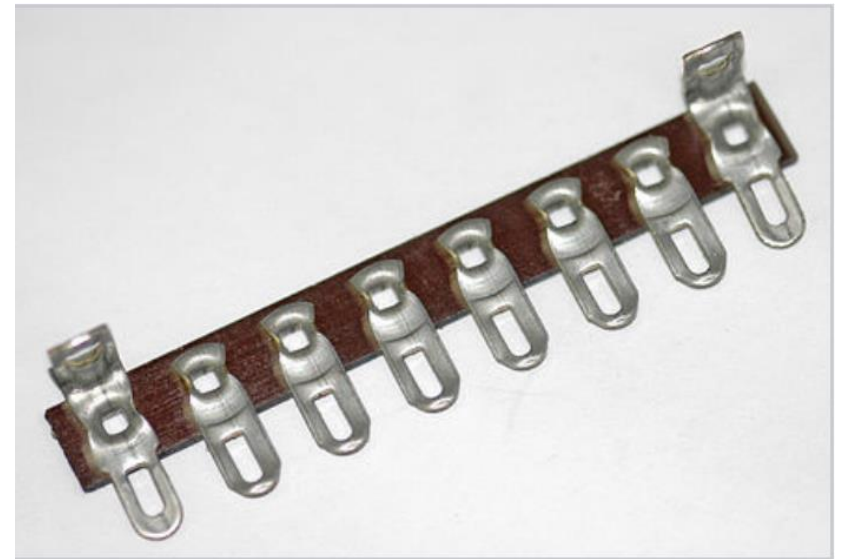
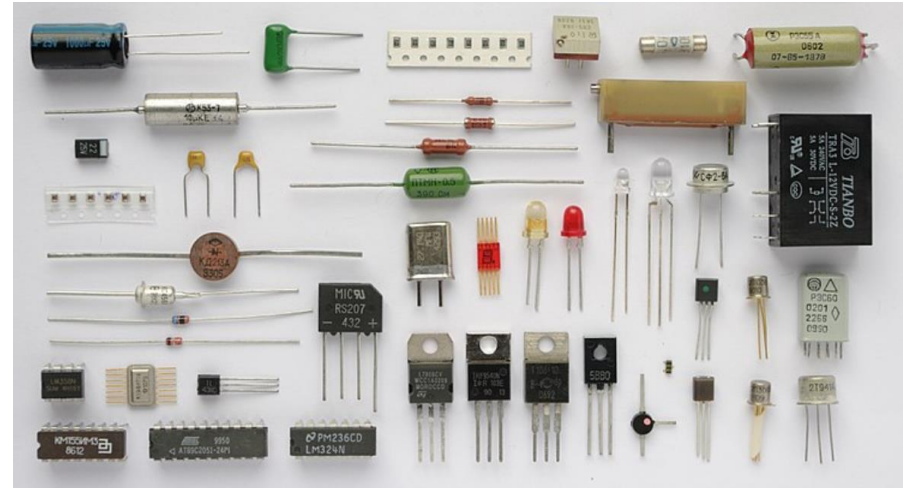
- Consistent self-discipline
- Functionality can be outdated by release time
- Change in requirements not in this release
- Prototype requirements ordering
- Huge cost to pay for attrition at end of release
- Tough to estimate amount of time for work and rework
- Feedback loop is too late

Domain disconnect - example

```
interface ElectricalObject {  
    Collection<IPin> getPins()  
}
```

```
class Splice implements ElectricalObject  
{...}
```

```
class Terminal implements ElectricalObject  
{...}
```



Agile methodologies

- Shorter release cycles
- Continuous integration and continuous delivery
- Feedback loop and reducing the time needed for feedback
- Backlog
- Return on Investment (ROI)
- It's a specification

Agile methodologies – stakeholders/actors

- Product owner
- Product marketing
- Scrum master
- Dev team
- QA team

Agile methodologies - backlog

- Initiatives

- Epics

- Stories

- Engineering tasks
 - QA tasks
 - Story bugs
 - Technical tasks
 - Marketing tasks

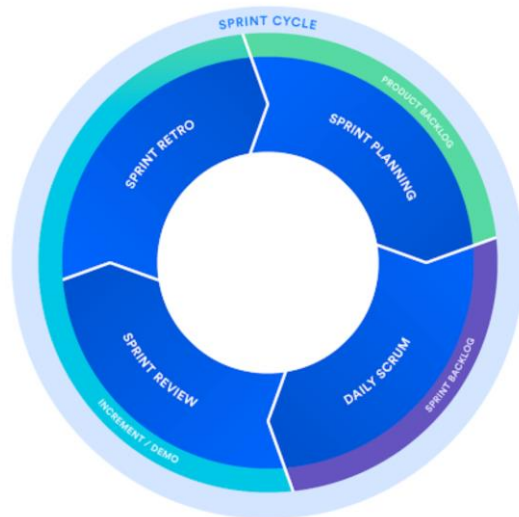
- Bugs

- Tasks

- Technical debts

Agile methodologies – ROI

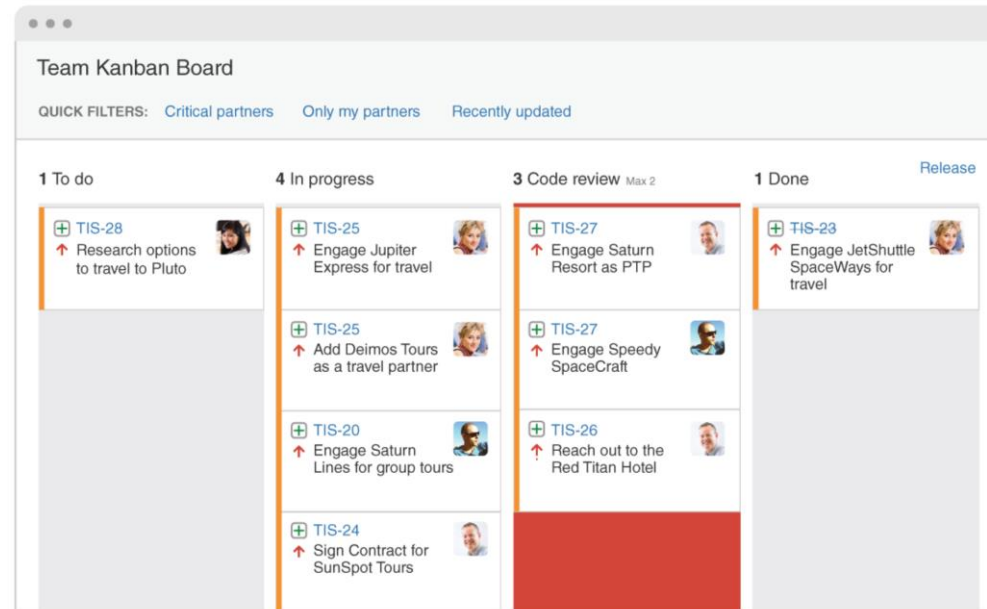
Scrum board



Time is fixed and better for commitment tracking

<https://www.atlassian.com/agile>

Kanban board



Flexible and followed for better tracking

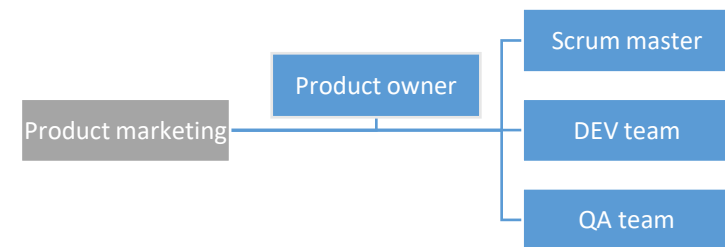
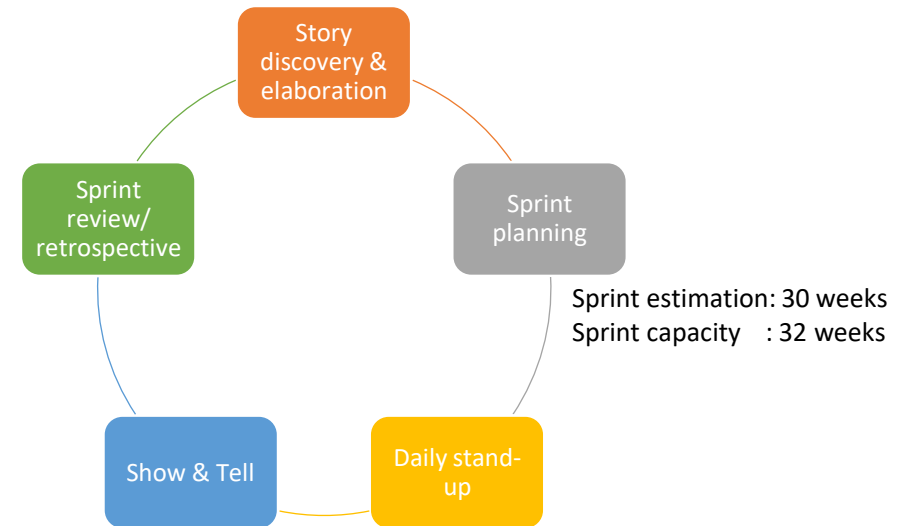
Agile methodologies – scrum board

- Story elaboration
 - Story discovery
 - Guestimates
- Sprint planning
 - Duration: typically 1 month or 2 weeks
 - Commitment to stakeholders
 - Work estimates for this duration
 - Planning poker
 - Fibonacci series based estimation – 1h, 2h, 3h, 5h, 8h or 1d, 13h, 21h (further breakdown)
- Daily scrum
 - Stand up team meeting for 15 minutes
 - Plan for the day
 - Blockages
- Show & Tell
 - Progress shown to stakeholders
 - Weekly typically
- Sprint review/retrospective
 - What's actually achieved
 - What went well and what can be improved
 - Celebration or small team activity

Agile methodologies – scrum board

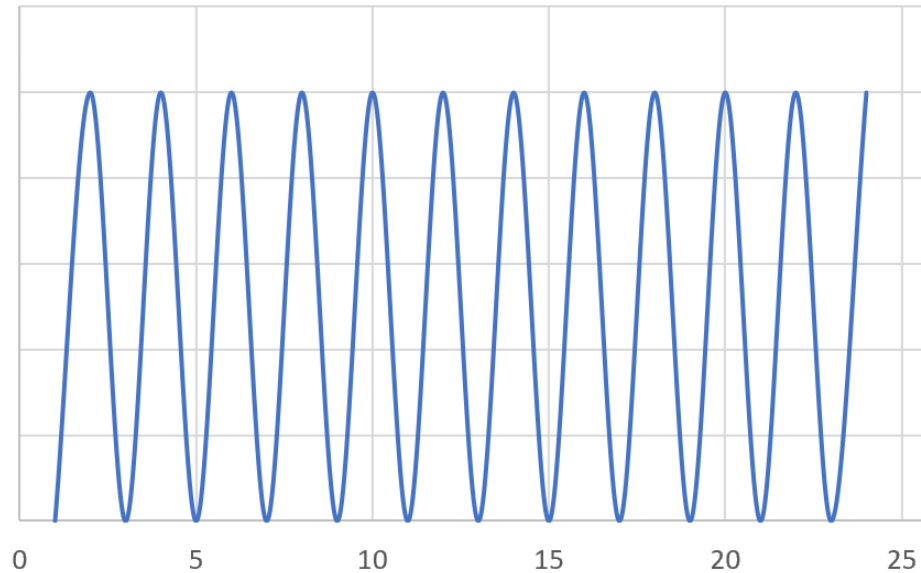
Release backlog – hypothetical e-commerce application

- Search – support top 20 results
- Search – provide ability to search based on name of product
- Search – provide category of products filter
- Search – support sorting based on relevance, recent, price or name
- Payment – support credit card payment
- Payment – support debit card payment
- Payment – support UPI payment (requirement came much later)
- Delivery – contactless delivery via OTP (requirement came much later)



Water fall model to Agile shift

Stress level



“Some amounts of stress are good to push you just to the level of optimal alertness, behavioural and cognitive performance.”

<https://news.berkeley.edu/2013/04/16/researchers-find-out-why-some-stress-is-good-for-you/>

Consistent self-discipline

- ✓ Imbibed in planning

Functionality can be outdated by release time

- ✓ Outdated functionality implemented in a Sprint is thrown away minimizing effort

Change in requirements not in this release

- ✓ Can be taken up as ROI changes equation

Prototype requirements ordering

- ✓ Localized to Sprint at hand. No need to prototype the uncertain functionality

Huge cost to pay for attrition at end of release

- ✓ Knowledge is spread across team

Tough to estimate amount of work and rework

- ✓ Entire team estimates and also gets fine tuned in retrospective meetings for effective estimation subsequently

Domain disconnect

- ✓ Gets detected earlier with stakeholders taking part in regular show & tell meetings

How about stress?

- ✓ Stress is indispensable but gets reduced down to the amount of exercise

Agile methodologies - summary

- Feedback loop and reducing the time needed for feedback
- Return on investment (ROI)
- Vertical slicing
- Smaller end to end work
- Increased accountability
- It's **not** about pushing hasty work so unaffordable legacy code

Books and references

[Lean software and devops](#)

https://en.wikipedia.org/wiki/Domain-driven_design

<http://xunitpatterns.com/>

[Refactoring to patterns](#)

[Leading Lean Software Development](#)

<http://www.poppendieck.com/reference.htm>

[Domain Driven Design - Tackling Complexity Software](#)



Agile methodologies – done

Questions?