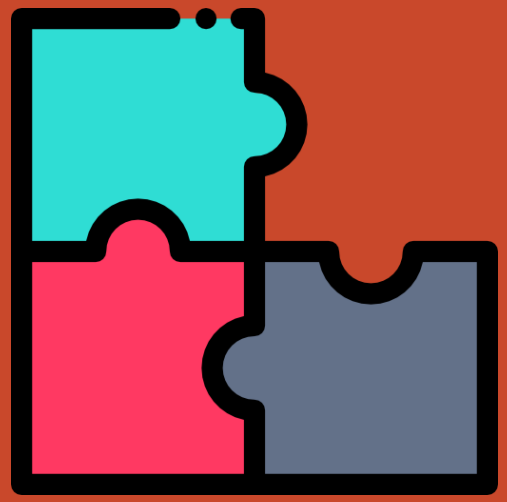
SOFTWARE ENGINEERING

Chapter 4.1: Agile Methodologies

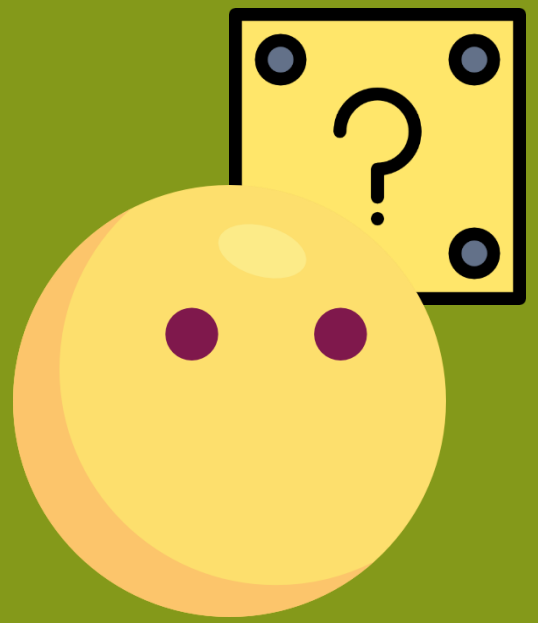
Part III

FDD Model

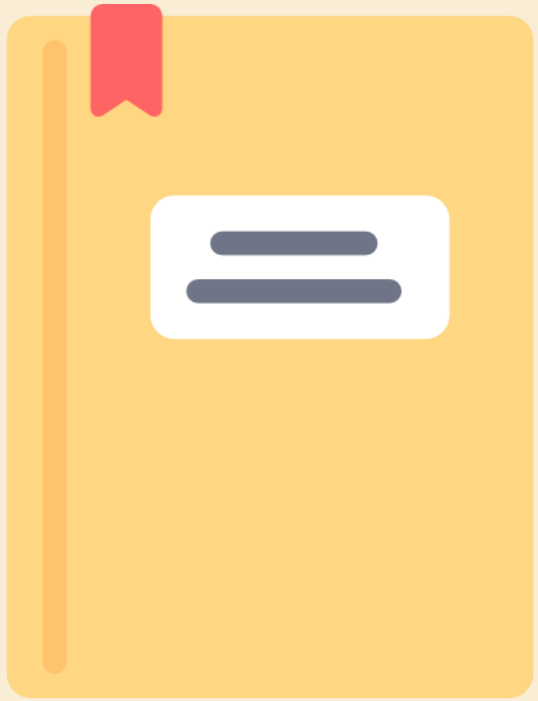
Feature Driven Development



What is it?



Definitions!



It’s an agile model that uses short iterations to develop functional software

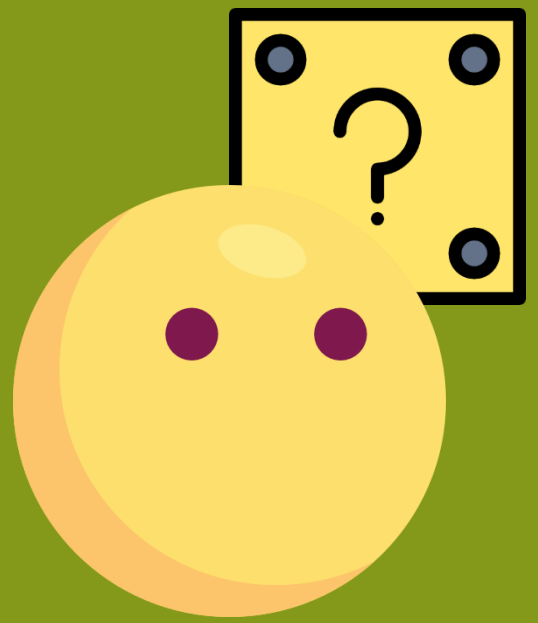
Firstly introduced in 1999 (Java modelling in color with UML)

Definitions!

Decomposes the project in Features which are valuable functions for the client

Remember ScheduleManager, Billing, Motion Capture, Online Payment on our examples...?

Something important first?



Roles!

Key Roles
Support Role
Additional Role

Project Manager

Chief Archtiect

Development Manager

Chief Programmer

Roles!

Key Roles
Support Role
Additional Role

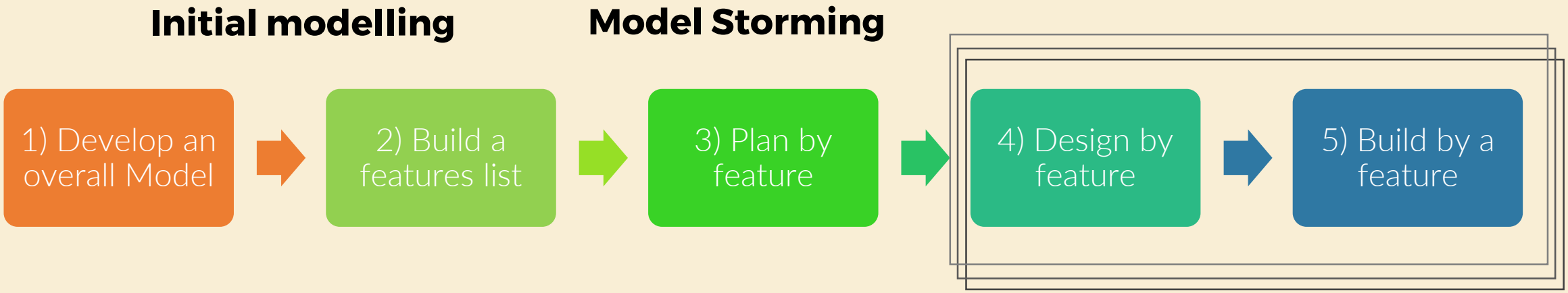
Class Owner

Domain Expert

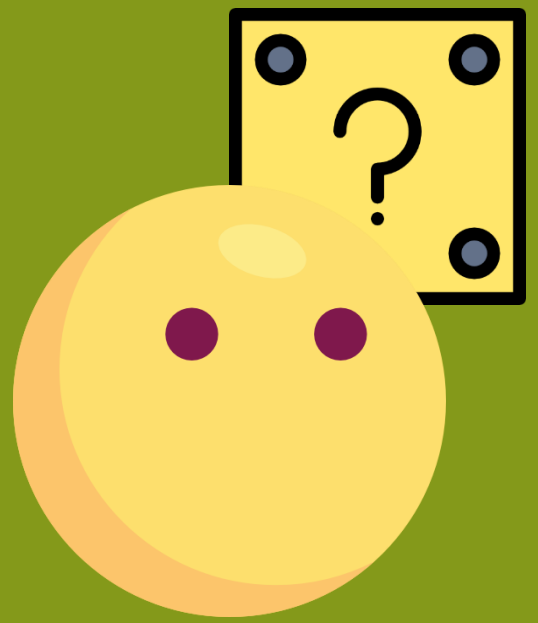
Feature Team

Show me the life-cycle diagram now!!

Here it is!!


Initial modelling
Model Storming
1) Develop an overall Model
2) Build a features list
3) Plan by feature
4) Design by feature
5) Build by a feature

Could you explain those phases?



1) Develop an overall Modell

Scope and context definition. Elicitaiton technique: Meetings

Multiple Object Models are developed!

Requirements engineering processes are not cleared at all!

2) Build a Feature List

Features are defined by all the domain areas

Each feature should be implemented in two weeks or less

The feature list is finally approved by the customer!

3) Plan by feature

Taking into account: Dependencies, Risks, Complexity and Workloads
Chief Programmers assign tasks to Class owners!

Project Manager

Development Manager

Chief Programmer

4) Design By Feature

Iterative activity. Lasts days or maximum two weeks

Sequence diagrams are fully elaborated

Design packages are reviewed and inspected

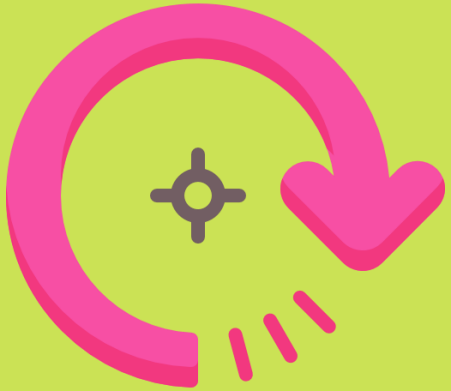
Class Owner

Chief Architect

Chief Programmer

5) Build by feature

Developing Iterations for features including:



Domain walkthrough, Design / Inspection, Code, Code Inspection

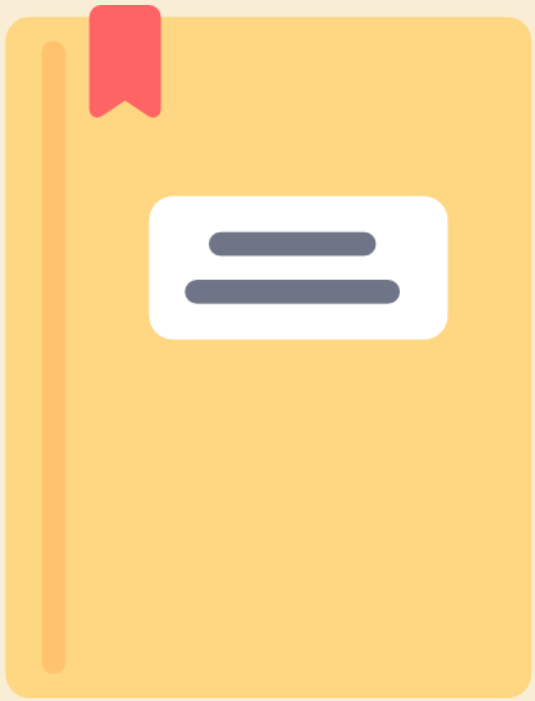
Unit testing, Integration Testing

Everyone has maximum two weeks to accomplish this activities!

Crystal Models

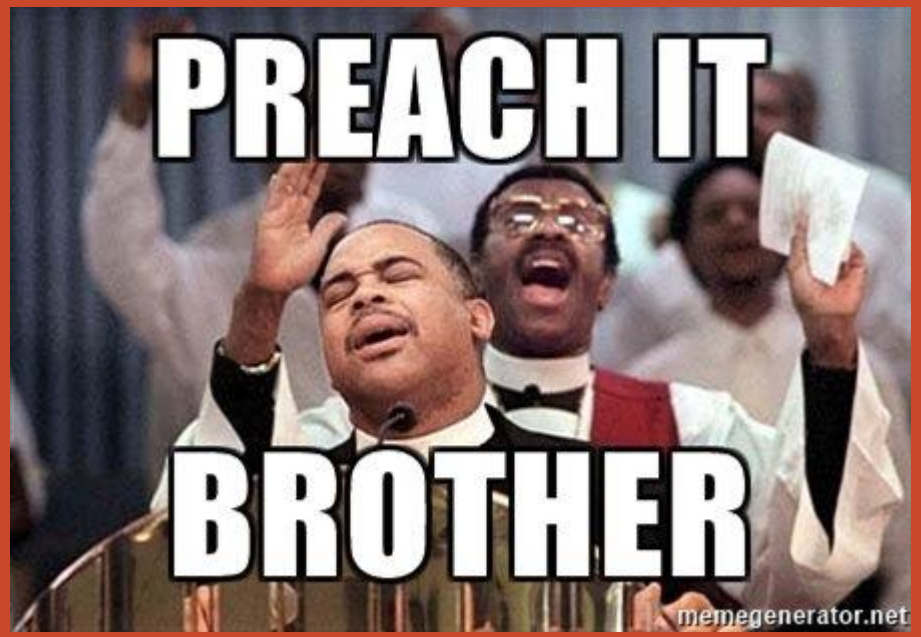
Clear, Yellow, Orange, Red

Definitions!



Crystal methods are a collection of software methodologies that can be used for different contexts [IBM, 1990]:

- Team size
- Criticality
- Complexity
- Number of People involved



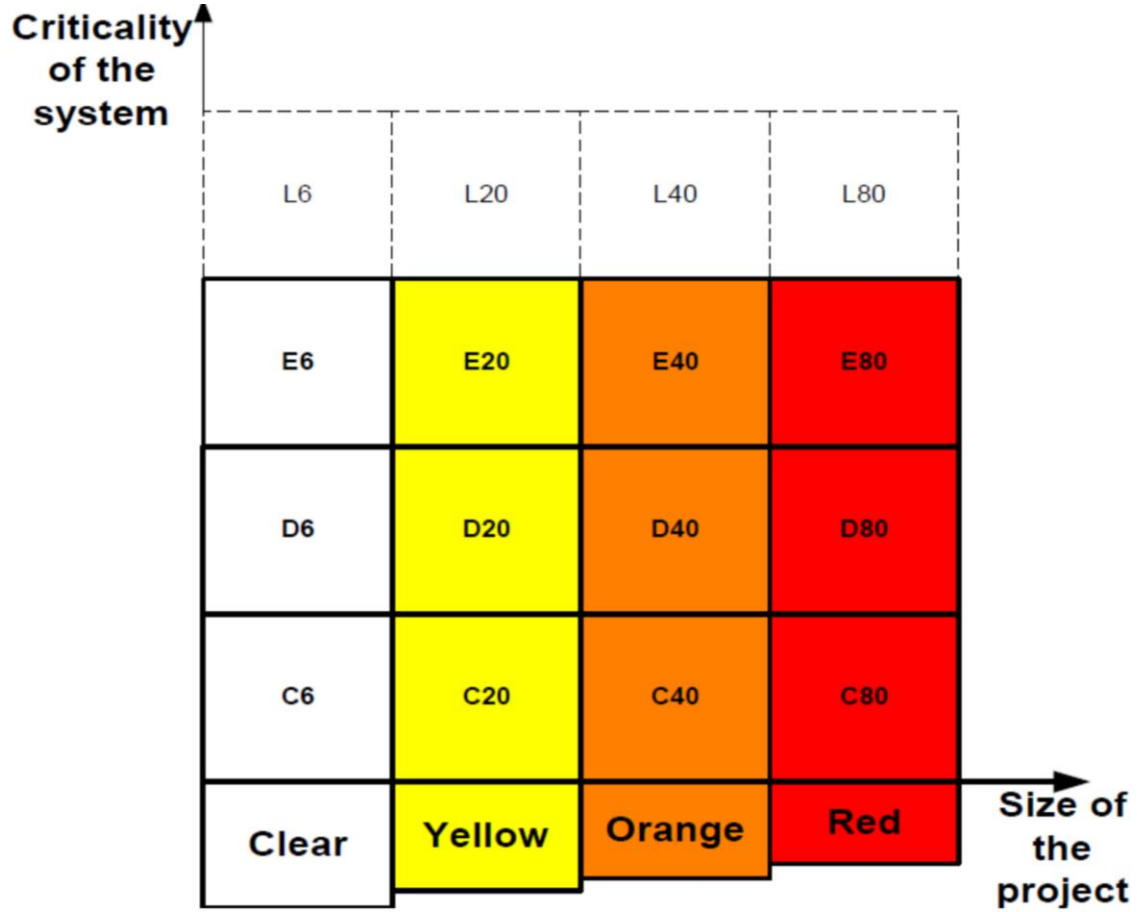
Fact!

“To the extent that you can replace the written documentation with face to face interaction, you can reduce the reliance on written ‘promissory’ notes and improve the likelihood of delivering the system”

Agile Software Development Ecosystems, 2002

Okay... Move on please!

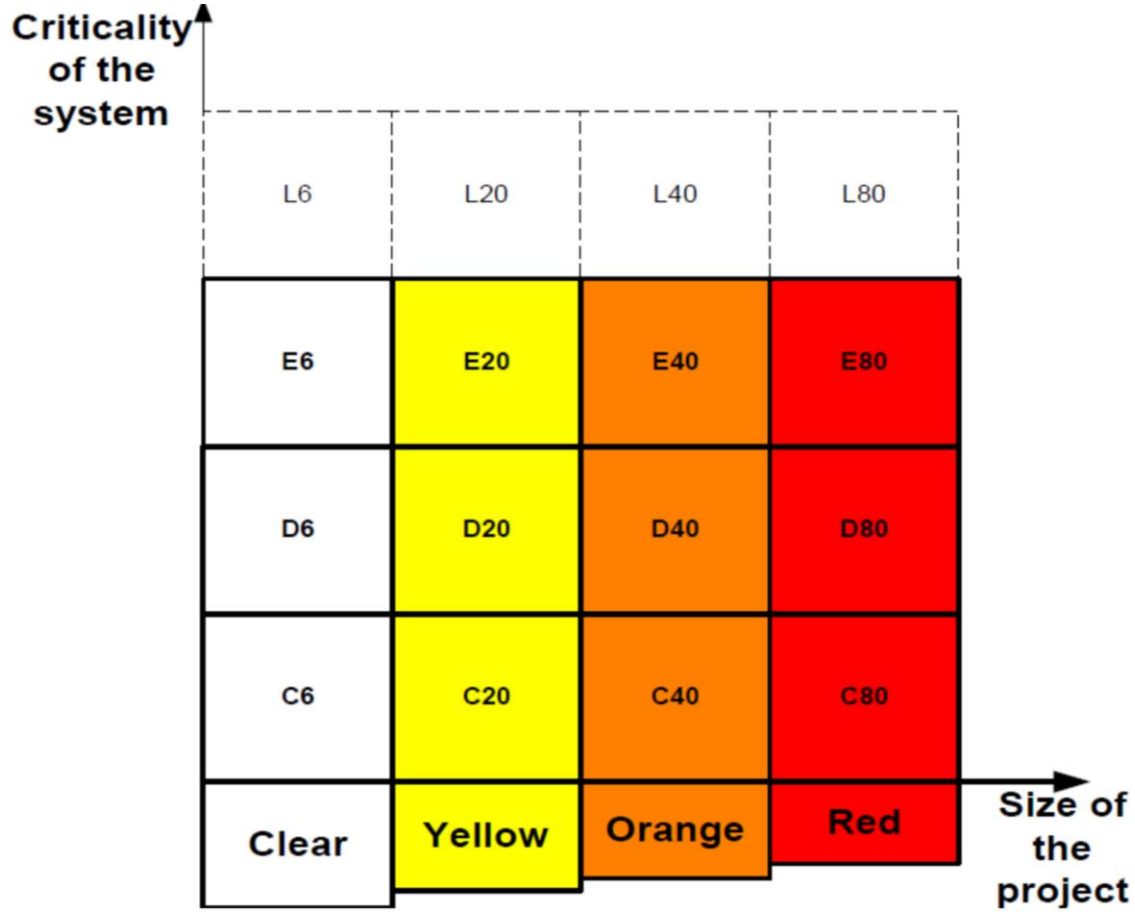
Crystal Methods – Family!



Use numbers to specify the team size for a given project! (6 members, 20 members, 40 members…)

Use caps lock characters to represent overall risk of the project!

Crystal Methods – Family!



Comfort

Discretionary Money

Essential Money

Life


Crystal Clear

Crystal Clear

- Suits best D6 Projects
- But it can also be used up to D10 or even E10 projects

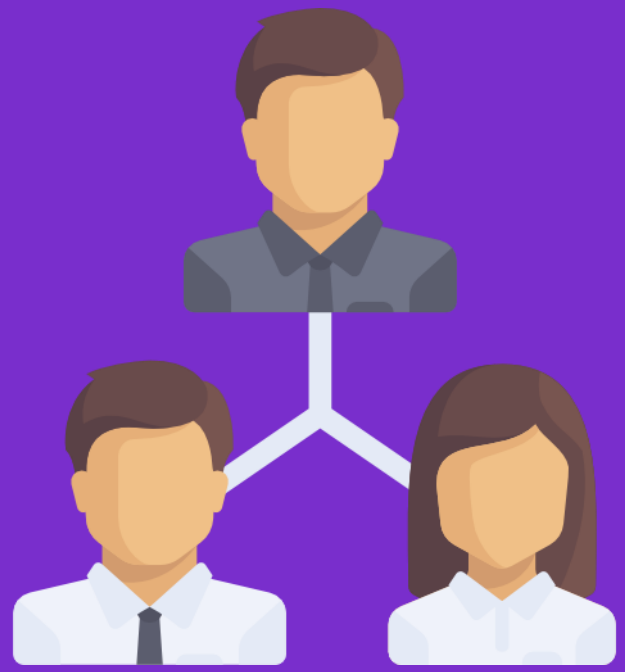
1) Frequent Deliverables!

Deliver tested code for real users! ASAP

2) Frequent Improvement

Teamwork is fundamental!!

If you’re bad, don’t worry about it… if you’re a beast you can still improve!



3) Osmotic Communication

Osmosis is always a passive way to learn things, team members should be located nearby so information can be picked up even when you're not actually participating in the conversation!

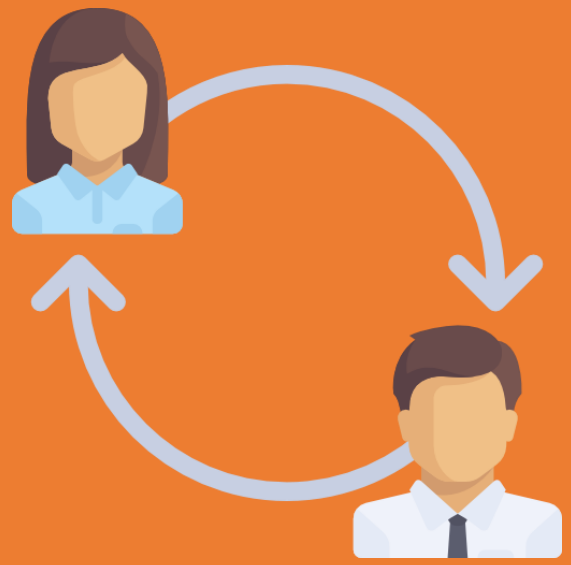
And... The optional ones... ???

4) Personal Safety

Every team member must be able to speak up!, shame or fear of ridicule is totally forbidden for this methodology

5) Focus

Knowing what to work on and being able to get on and do it. This suggests clear communication, prioritisation of requirements, goal setting, etc. It also means reducing context switching.



6) *Easy Access to Expert Users*

Feedback from real users is invaluable!

7) Technical Environment with Automated Tests, Configuration Management & Frequent Integration

Nothing to say here...!

Crystal Clear Roles!

Sponsor

Senior Designer

Unit Tester

Senior Programmer

Crystal Clear Roles!

Programmer

Business Expert

Crystal Orange

Crystal Orange

- Suits best D40 Projects.
- If tuned fine you can scale the risk up to E

Just create ‘Rings’ (Remember them?)
And… Roles… Add more roles!

Crystal Orange Additional Roles!

UI Designer

Technical Facilitator

Database designer

Business Analyst

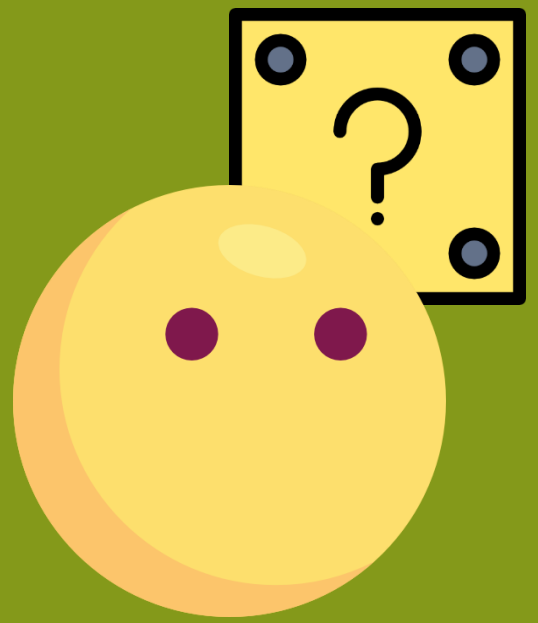
Crystal Clear Additional Roles!

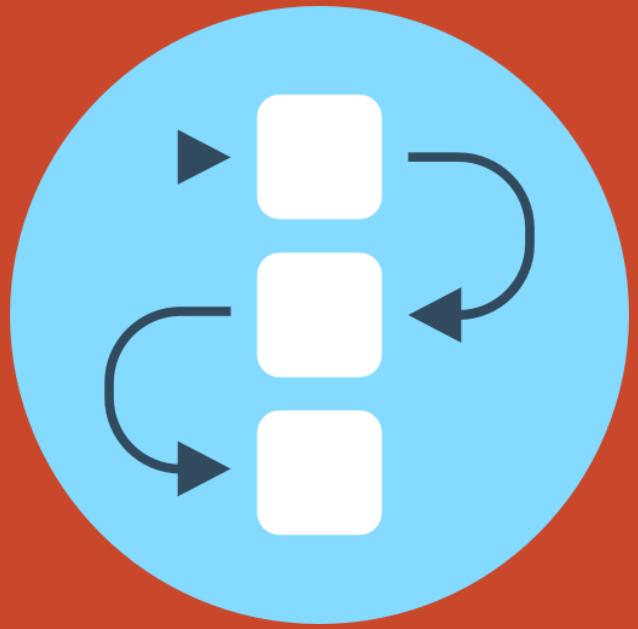
Usage analyst

Tester

Writer

What about Kanban?





Kanban!

Is A way to organize the chaos that surrounds so many delivery teams by making the need for prioritization and focus clear.

Start what you do now!

- Kanban can be used in any process model. It’s just a way to manage things and focus on what’s important.

- Encourages the small incremental changes on the system while it’s being developed.

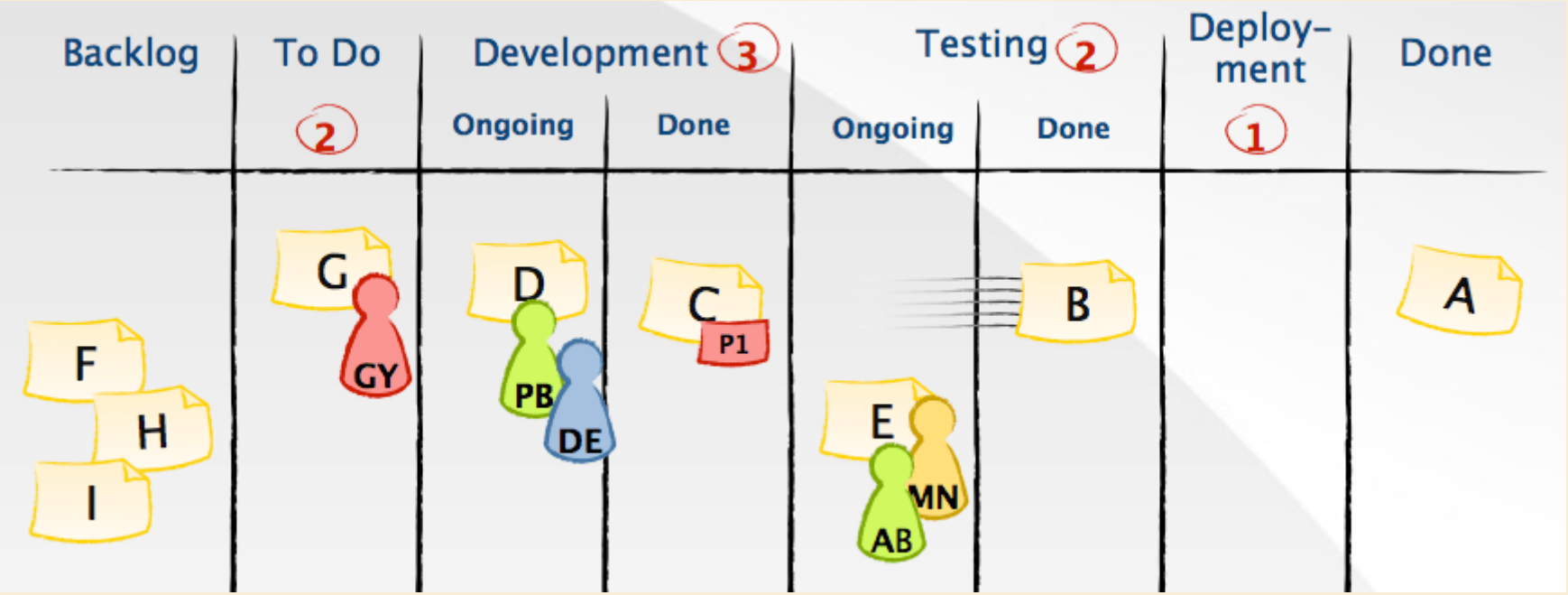
The same principles...

Everyone can lead!...

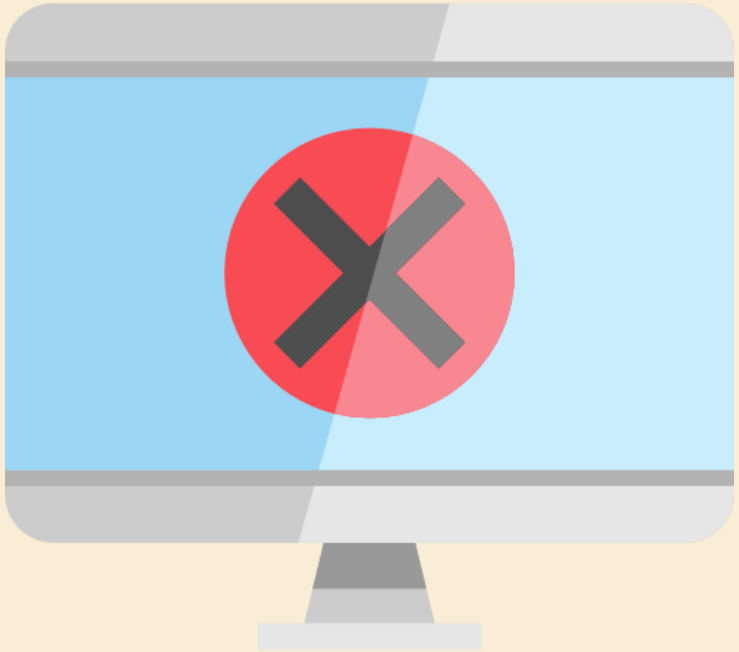
Trust your team!...

Continual Improvements!...

Visualize things!



Delays



Delays can be measured in this 4 cathegories:

- Expedite
- Fixed Date
- Standard
- Intangible

Project Changers

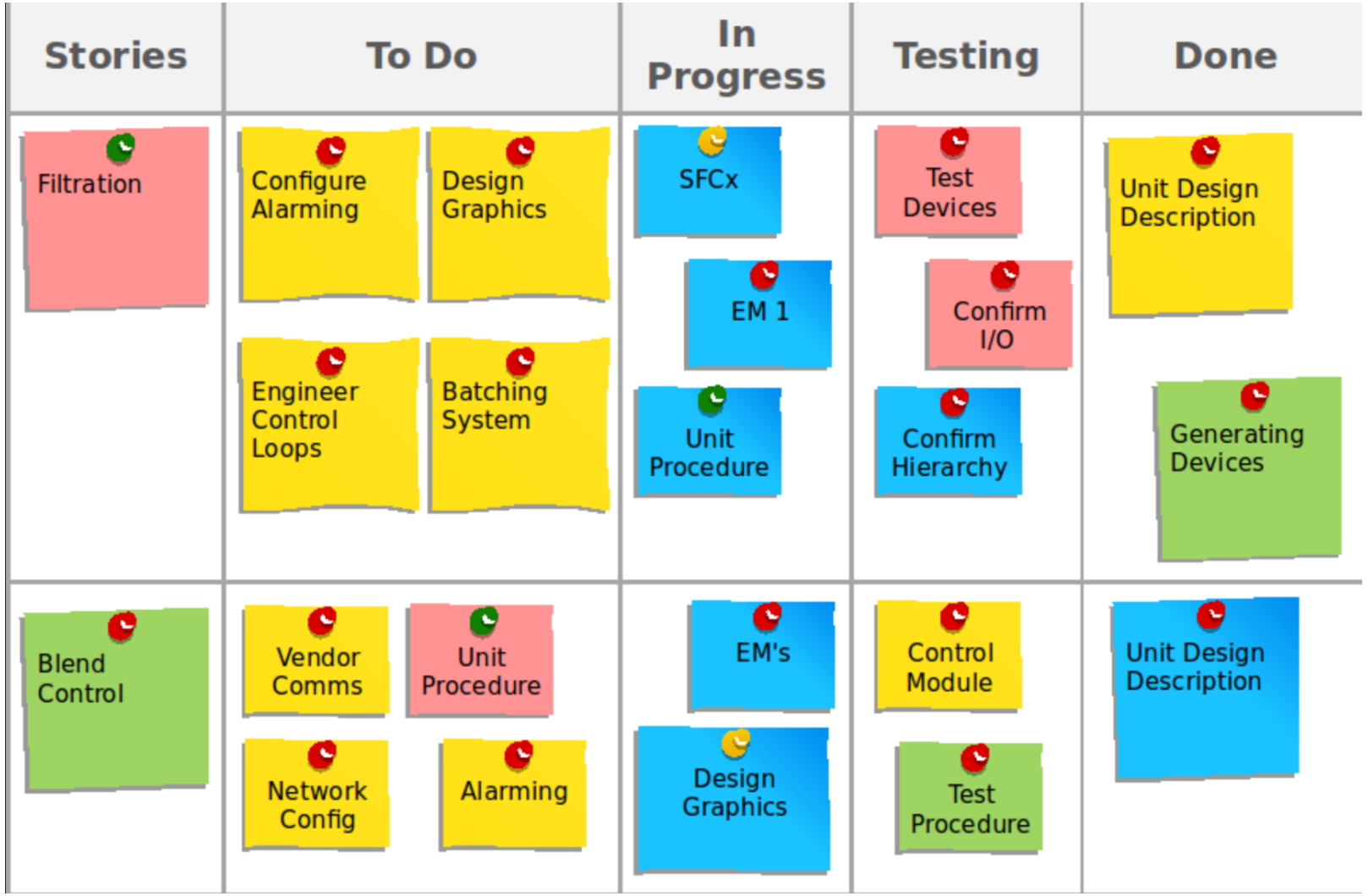
Business Changers:

- Differentiators
- Spoilers
- Cost reducers
- Table stakes

More principles!!

- Communicate everything. Get related to every activity in the flow!
- Discuss activities and perform meetings in pro of continuous learning.

Another Example!



References

- [STAPLETON] Jennifer Stapleton *DSDM Business Focused Development.*
- [SOMMERVILLE] Ian Sommervile. *Software Engineering 9th Edition*
- [SCHMIDT] Richard Schmidt. *Software Development Architecture-Driven Software Development*
- [STEPHENS] Beginning Software Engineering. 2015
- [CROOKSHANKS] Software Development Techniques. 2015

Class has died… for today!