SOFTWARE ENGINEERING

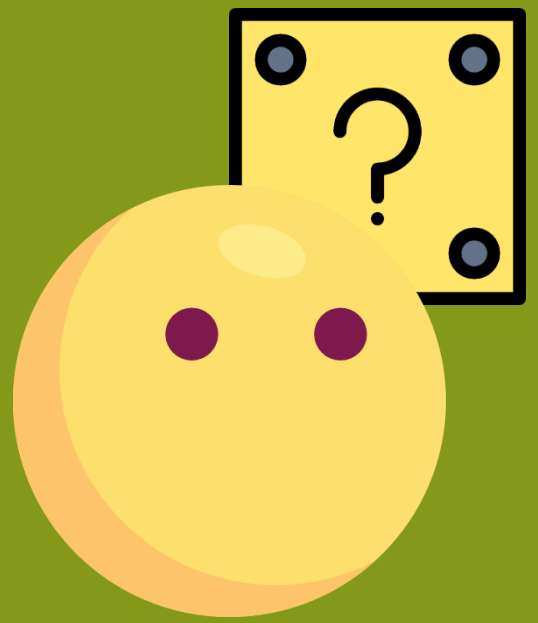
Chapter 4.1: Agile Methodologies

Part II

DSDM Methodology

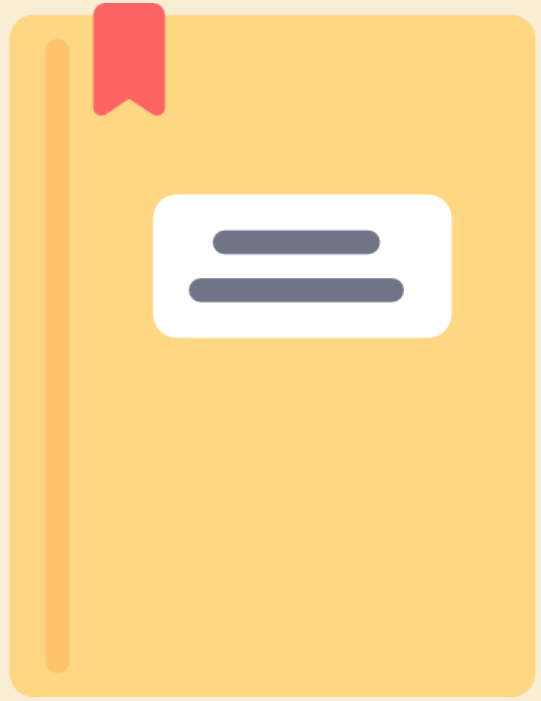
Dynamic Systems Developed Method

What is it?



Definitions!

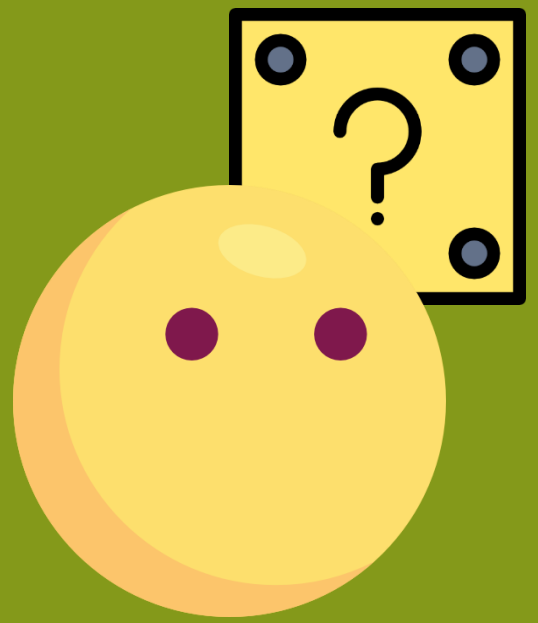
The Dynamic Systems Development Method is a framework which embodies much of the current knowledge about project management [STAPLETON]



Definitions!

It’s the convergence between software development, software engineering and business modelling

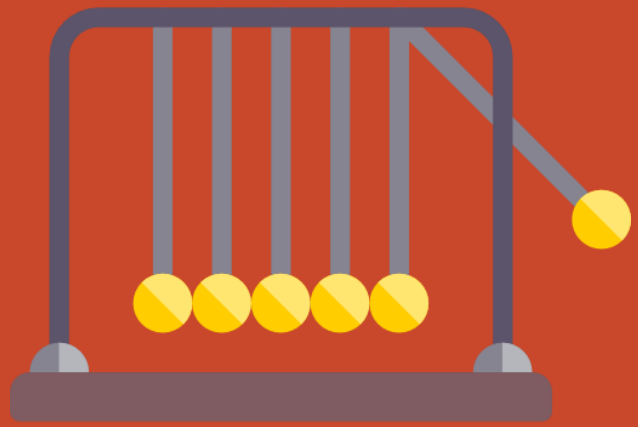
Does it have principles...?



1) Active user Involvement is Imperative

Reduces costs!, a set of users is required to keep the project going!

2) Teams Must be Empowered to Make Decisions



Tensions and misscomunications need to be deleted from the team members!

Decision Making!

- Requirements in practice
- Which functionality needs to be in a given increment
- Prioritization of requirements and features
- Fine details of the technical solution

3) Frequent Deliverables!

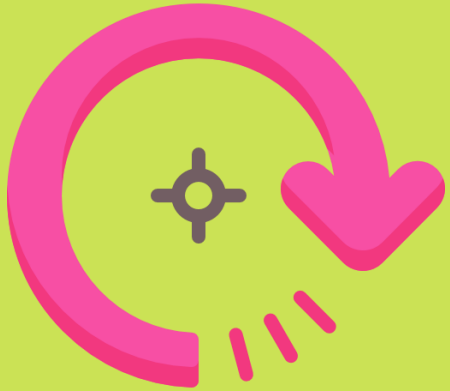
Things need to be delivered so that, errors can be corrected shortly!

4) Fitness for Business is Criterion for Accepted Deliverables

Refactoring!, Deep understanding of patterns and architecture

5) Iterative and Incremental Development is Mandatory

As XP, software development is decomposed by small iterations!



6) All Changes must be reversible

Iterations are small increments of development, and who knows when the client needs to take things backwards!

7) Requirements are Baselined at High-Level

Not all the requirements are going to change, we need to freeze the highest level ones!!

8) Testing is Integrated Throughout the Lifecycle

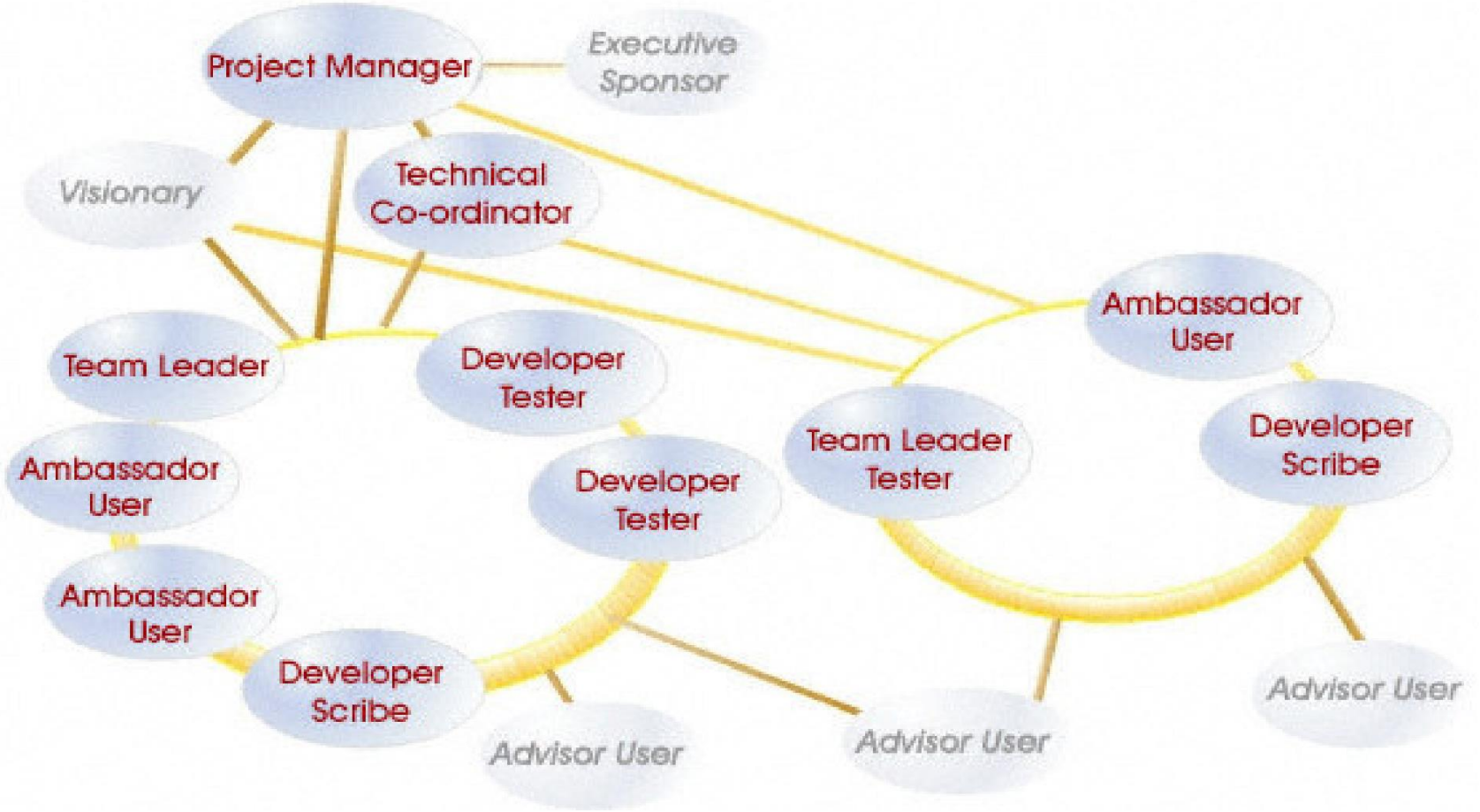
You need to test throughout all the development process!, even if those tests are small documents or surveys..., everything eneds to be tested ASAP!

9) Collaboration and Co-Operation

Technical staff and business staff will work together!!

I said!

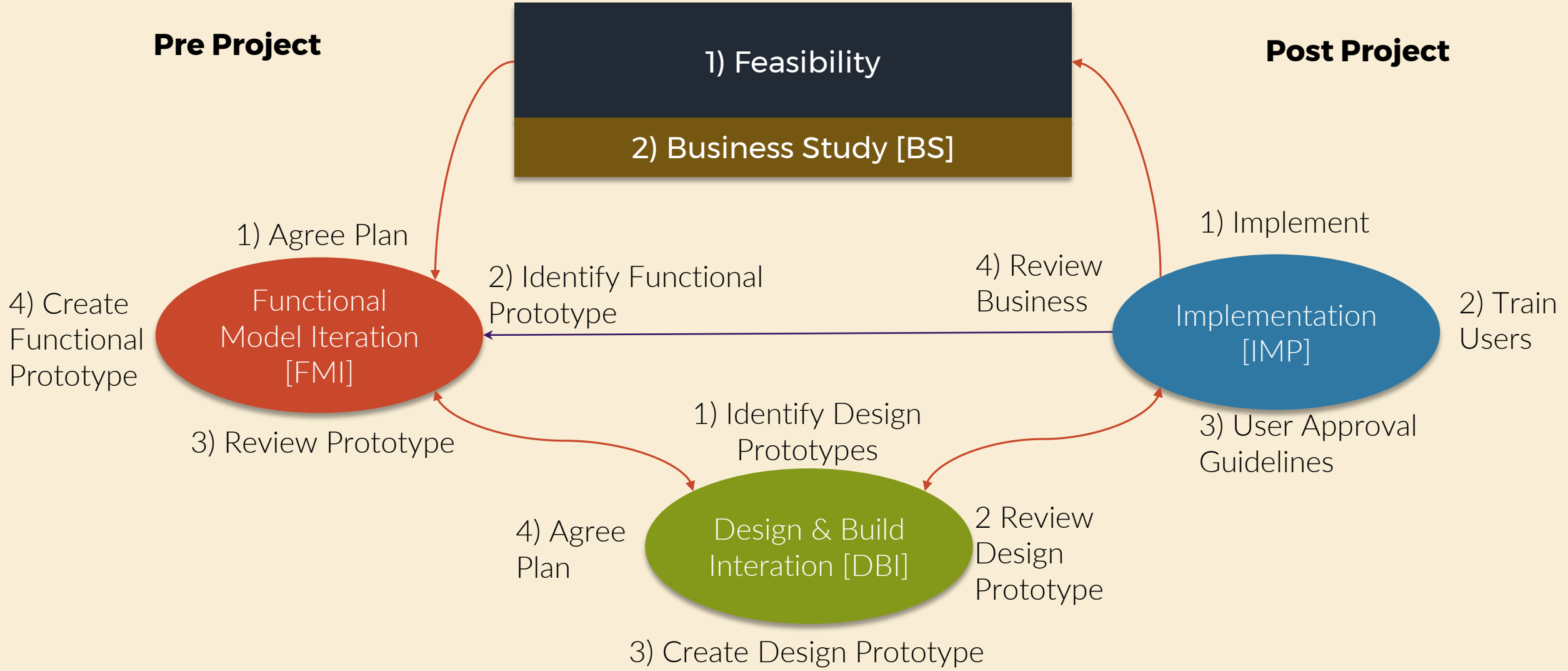
Roles on DSDM



Relationships of Project Roles - Figure 1

[DSDM04a]

WBS of the DSDM Methodology

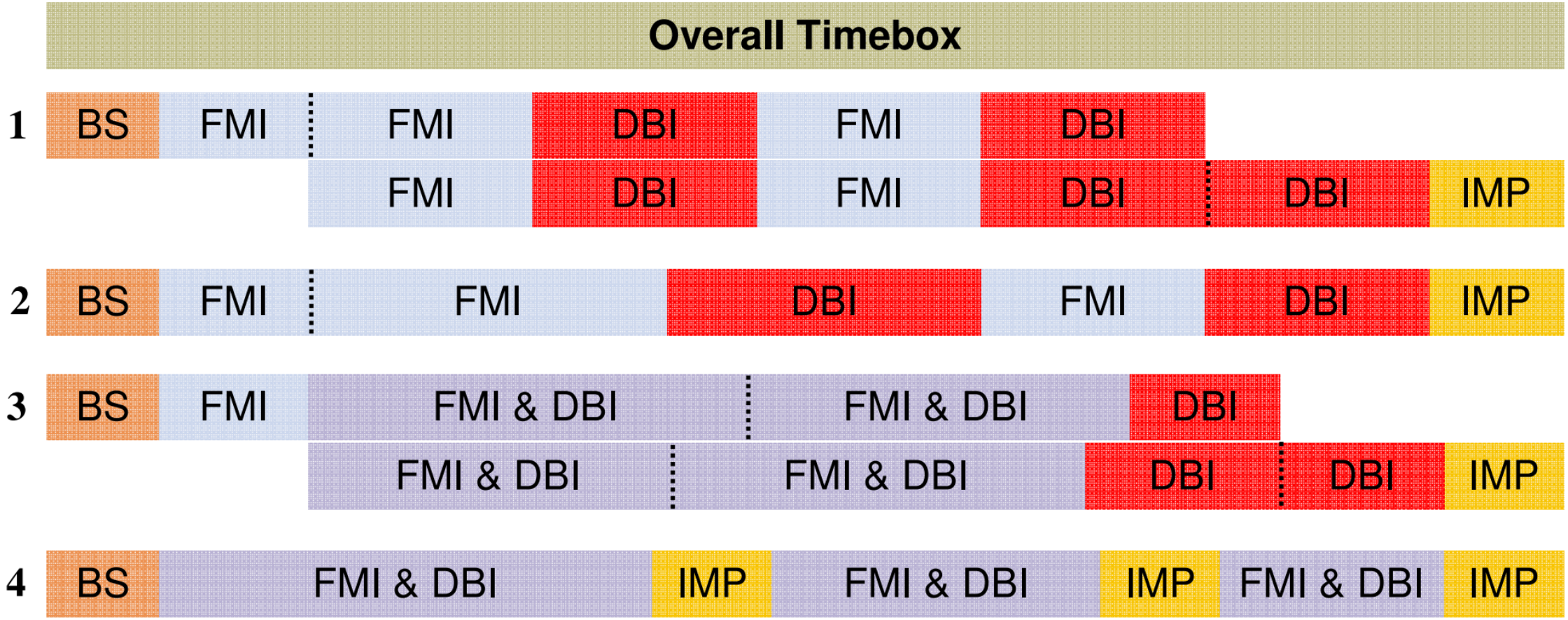


TimeBoxing!

- Milestones where helpful the last time! But now timeboxes can help us to estimate times and tasks!

- A timebox is an interval no longer than 6 weeks and can have several activities!

Timeboxing example!



Timebox Examples - Figure 3

[DSDM04a]

MoSCoW Rule

1. **Must have:** Mandatory features!!!
2. **Should Have:** Important features but can be omitted if risks happen!
3. **Could Have:** Enhancement features that add value to the system!
4. **Want to Have:** Valuable features for few users (low value overall)

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!