Week1:

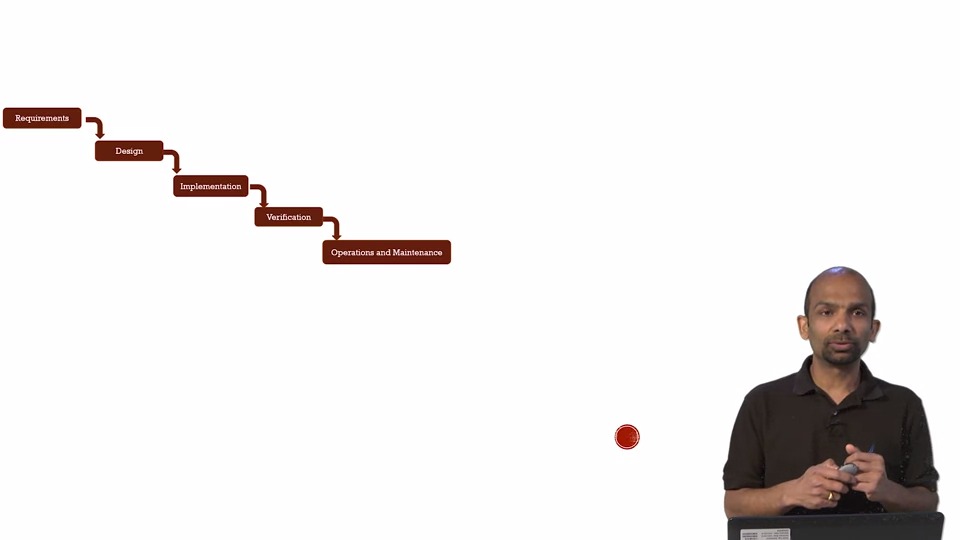
Estimated time to complete: 3 hours

Actual time:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Start Time | End Time | Accomplishment | Time(min) |
| 7/01/2020 | 12:06 | 12:20 | 2 video completed. 3rd ongoing. | 14 |
| 7/01/2020 | 1:15 |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Introduction to course:

What software development looks like?

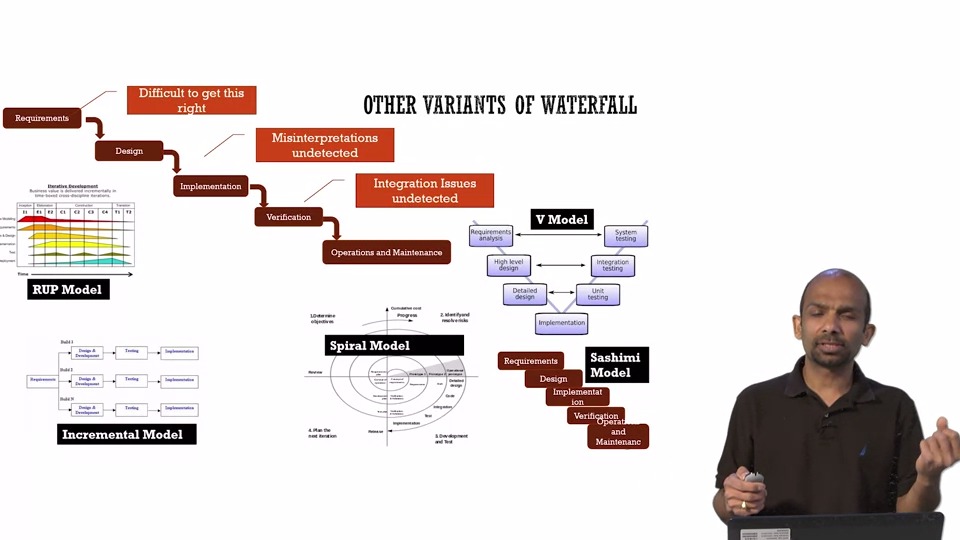


The model that we just reviewed is called, a waterfall method,

where you go from phase to phase to phase.

Requirements, design, implementation.

Img2

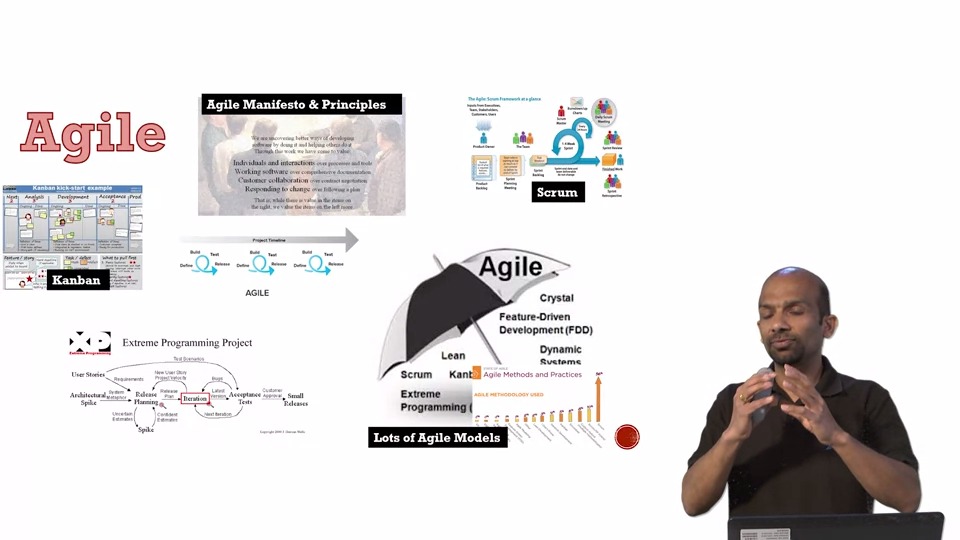


As all of these variants were coming along,

there was another thought process that were

evolving in the software industry, called Agile.

So, Agile is not a model. It's a mindset.

Img3 

The basic idea behind all of this model,

was that instead of building this whole one year cycle,

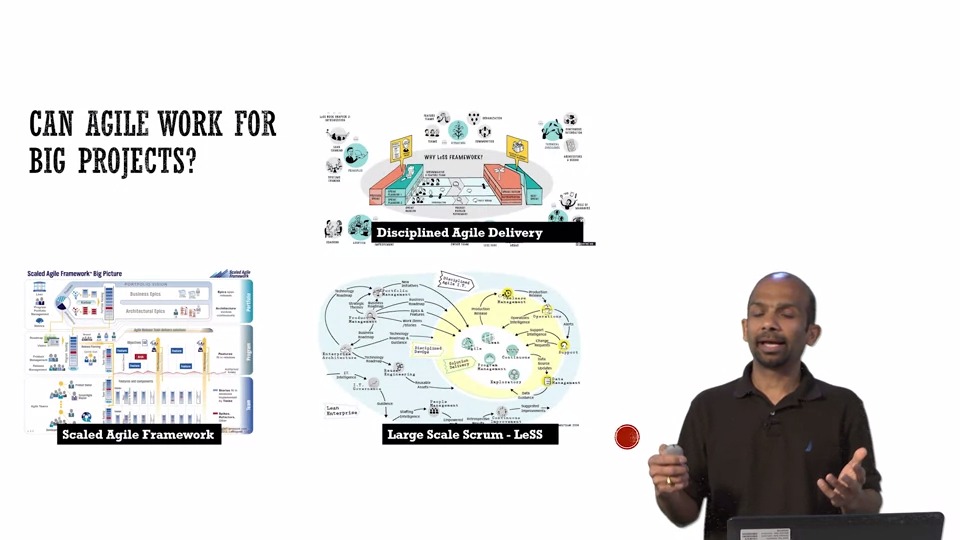
you build in short cycles.

So, you define little,

you build, you test, and you learn.

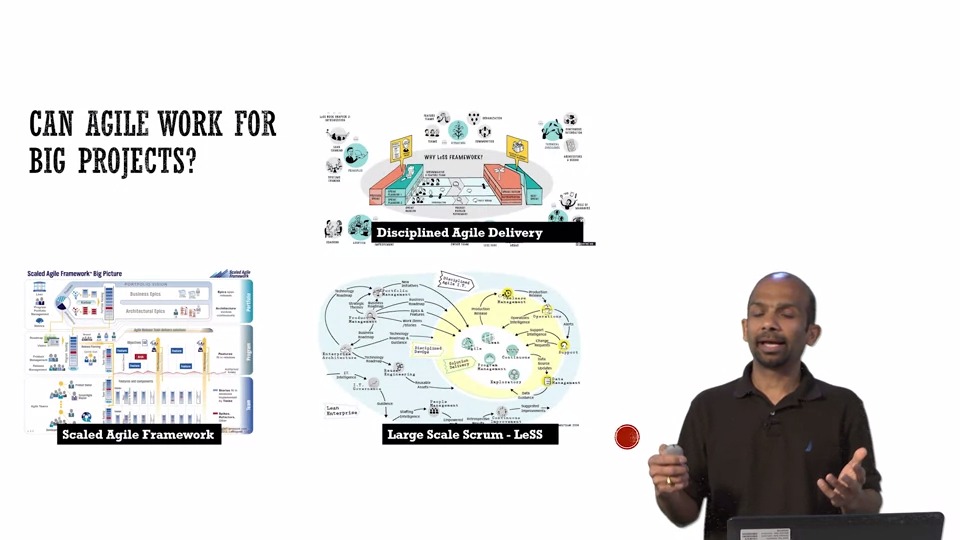
And then again, you just keep doing this.

That way you can adjust to the market quickly.

Img4

They march towards a common goal.

And, that led to a culture of DevOps.

Img 5

Now, as this model of Agile was getting popular in industry in a smaller scale,

people started to wonder,

"Can we apply this on a bigger projects?"

And hence, many new models evolve,

like Scale Agile Framework,

Large Scale Scrum, and Disciplined Agile Framework.

Img6

they wanted to learn if there is a cheaper and a faster way to learn.

And, the concepts like Lean Startup and Design Thinking came into being,

where, you find the cheapest way to learn the one cycle.

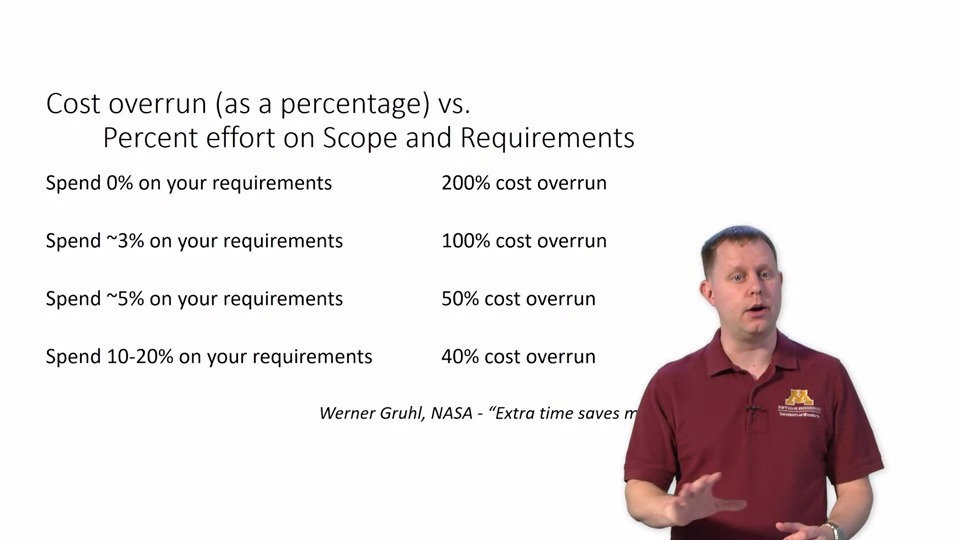
What is Requirement Specification?

-Process

-Product of process

Importance of Requirement Specification:

-Engineering Argument



-Economic Argument: Problem found after deployment cost 1000 times more to solve than the problem found during staging area.

Requirement vs Specification:

-User Requirement:

Requirement is for user or client.

Write in user language

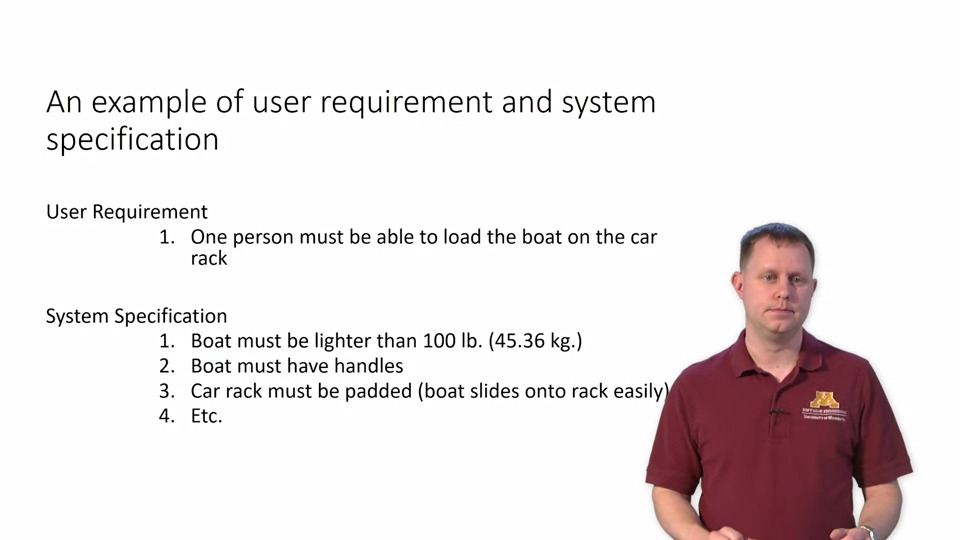
-System Specification:

System spec is the usually more precise or constraining statement of how the system will meet the user requirements.

Write in system language.

Specification is for Developing team.

Example:



Non functional Requirement:

Requirements which don't specify what the system will do, but rather how the system will perform the behaviors.

Properties:

Define system properties and constraints.

-process requirement: Process requirements may be things like mandating a particular case system, that is, a computer-assisted software engineering tool like Microsoft Project or Ajira, a bug-tracking software.

-more critical than functional requirements

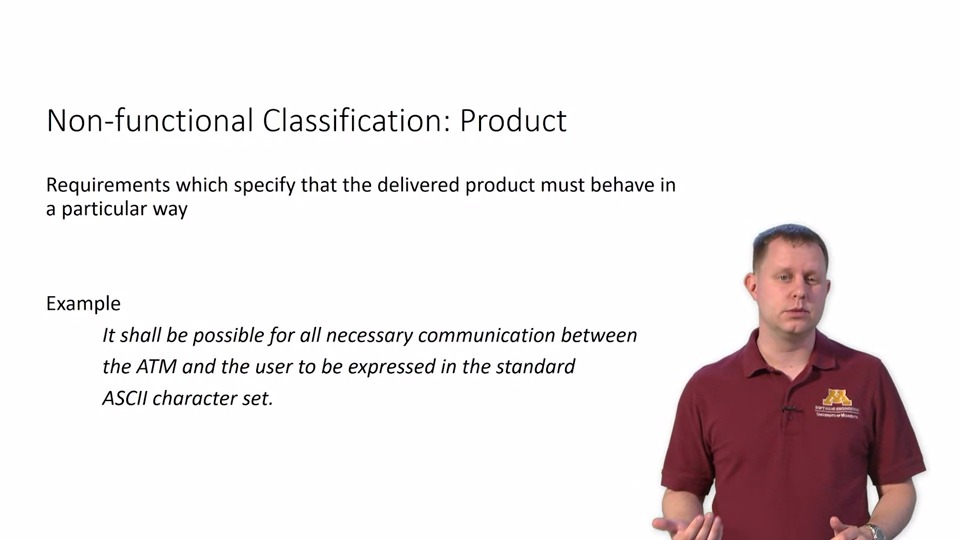
Classification:

-Product Requirement

-Organization Requirement

-External Requirement

Product Requirement:



Product requirements which are non-functional talk about specific

behavior.

This is often in the form of protocol requirements, encodings, or

encryption requirements, that sort of thing.

They are requirements on the product itself.

They are requirements on the product itself.

Organizational requirements:



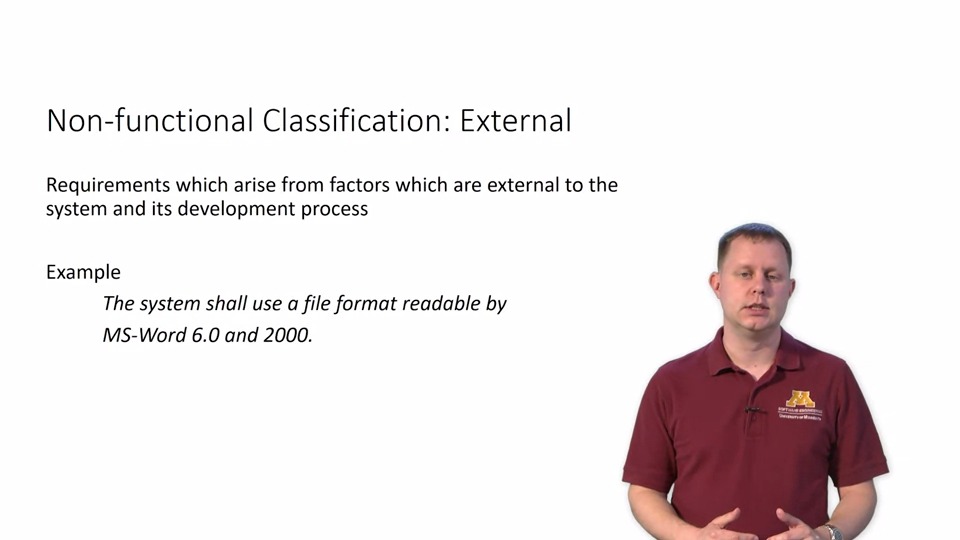
Organizational requirements are those that are defined by the company.

Company standards, your development team's code style requirements,

even the development process itself like using SCRAM

could be defined as something like this.

External Requirement:



And then external constraints are a big factor,

especially in regulated industries.

When the FAA says you have to use this development process or

meet these code coverage testing metrics, that's all there is to it.

You have to.

And if they change, you have to change.

This is the kind of control we're talking about,

being impacted by external entities.

That's something you want to document early on in the process.

WRSPM Refrence Model (world machine model) :