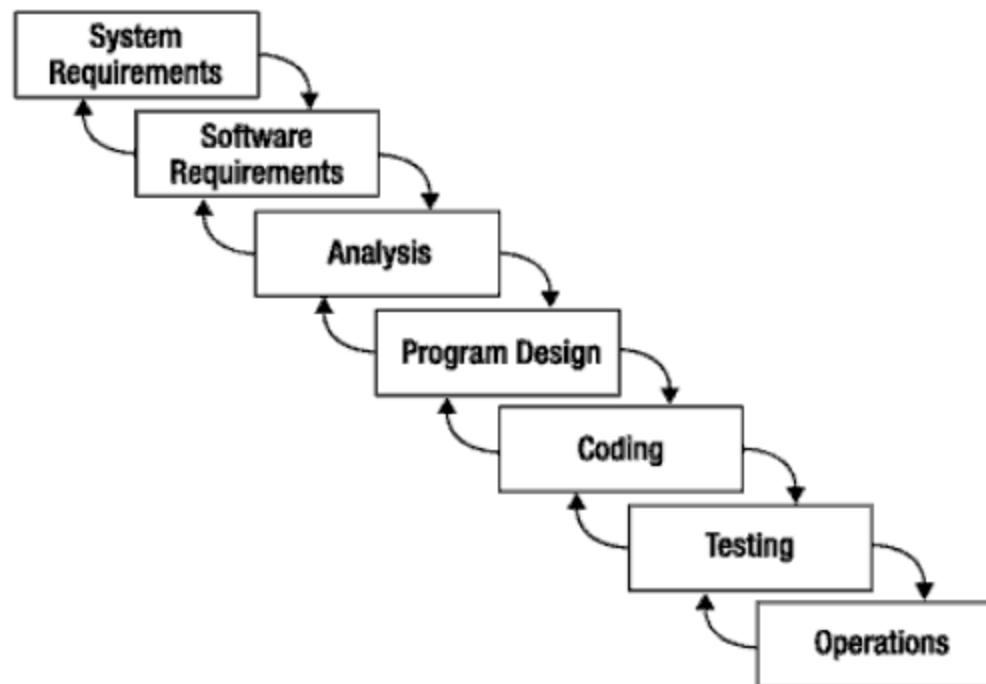


Intro to Scrum

What is Scrum?

- An answer to traditional “fixed cost / strict requirements” contracts which had very high rates of failure
- Recognizes the untenability of the “waterfall” method of development



What is Scrum?

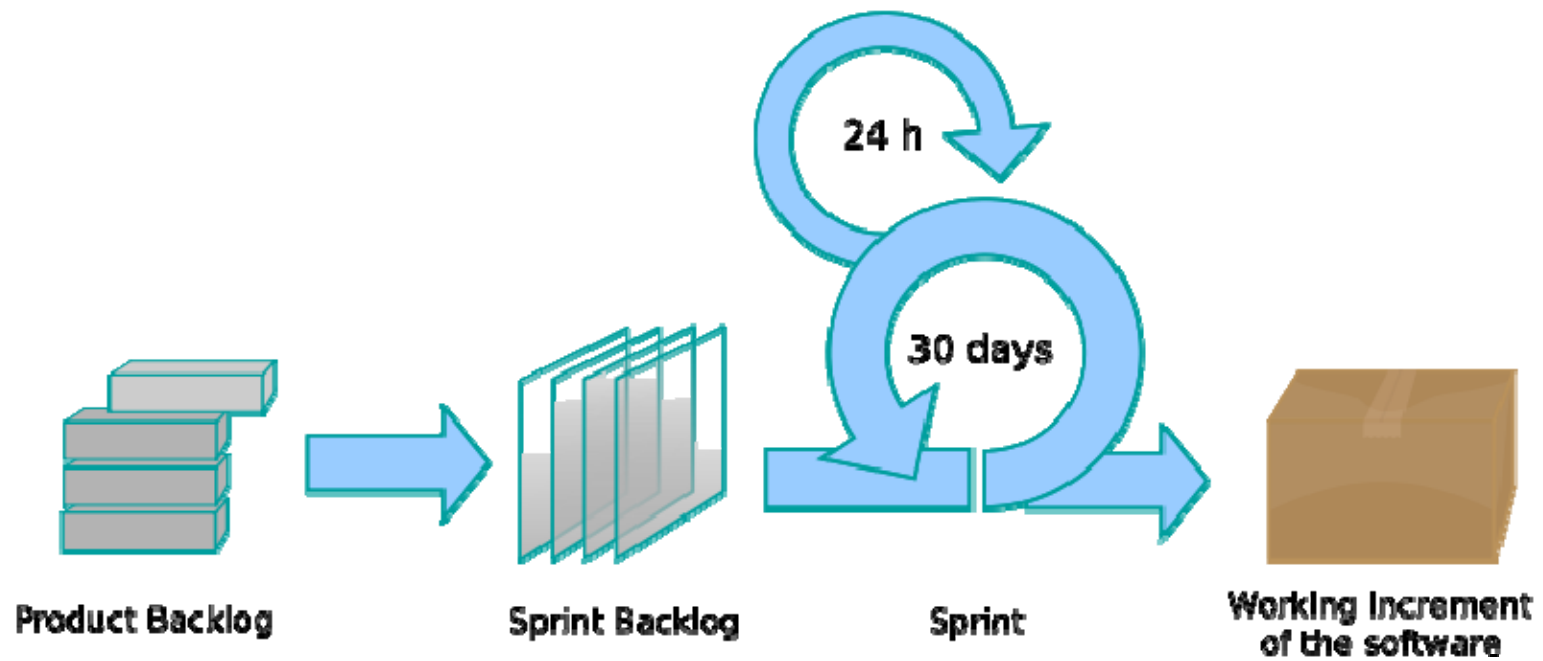
- An “Agile” process
 - ***Nimble*** – requirements & priorities change; embrace it!
 - ***Synergistic*** – get clients and developers working together, not against each other
 - ***Empowering*** – let those who are closest to the work make the granular decisions
- Agile manifesto:
 - ✓ Individuals and interactions over processes and tools
 - ✓ Working software over comprehensive documentation
 - ✓ Customer collaboration over contract negotiation
 - ✓ Responding to change over following a plan

That is, while there is value in the items on the right,
we value the items on the left more.

What is Scrum?

Scrum is a process skeleton that includes a set of practices and predefined roles. The main roles in Scrum are the ***ScrumMaster*** who maintains the processes and works similar to a project manager, the ***Product Owner*** who represents the stakeholders, and the ***Team*** which includes the developers.

Scrum process flow



Scrum Roles

Am I a pig or a chicken?

A pig and a chicken are walking down a road. The chicken looks at the pig and says,

"Hey, why don't we open a restaurant?" The pig looks back at the chicken and says,

"Good idea, what do you want to call it?" The chicken thinks about it and says,

"Why don't we call it 'Ham and Eggs'?"

"I don't think so," says the pig,

"I'd be ***committed*** but you'd only be ***involved***."

Scrum Roles

Am I a pig or a chicken?

Moral of the story:

- *pigs* are committed to building software regularly and frequently
- everyone else is a *chicken*: interested in the project but really irrelevant because if it fails they're not a *pig*, that is they weren't the ones that committed to doing it.
- The needs, desires, ideas and influences of the *chicken* roles are taken into account, but not in any way letting it affect or distort or get in the way of the actual Scrum project.

Scrum Roles – “Pigs”

Pigs are the ones committed to the project and the Scrum process; they are the ones with “their bacon on the line.”

- **Product owner**

The Product Owner represents the voice of the customer. They ensure that the Scrum Team works with the right things from a business perspective. The Product Owner writes User Stories, prioritizes them, then places them in the Product Backlog.

- **ScrumMaster (or project “facilitator”)**

Scrum is facilitated by a ScrumMaster, whose primary job is to remove impediments to the ability of the team to deliver the sprint goal. The ScrumMaster acts as a buffer between the team and any distracting influences. The ScrumMaster ensures that the Scrum process is used as intended. The ScrumMaster is the enforcer of rules.

- **Team**

The team has the responsibility to deliver the product. A small team with cross-functional skills to do the actual work (designer, developer etc.).

Scrum Roles – “Chickens”

Chicken roles are not part of the actual Scrum process, but must be taken into account. It is important for these people to be engaged and provide feedback into the outputs for review and planning of each sprint.

- **Users**

The software is being built for someone! If software is not used - much like 'the tree falling in a forest' riddle - was it ever written?

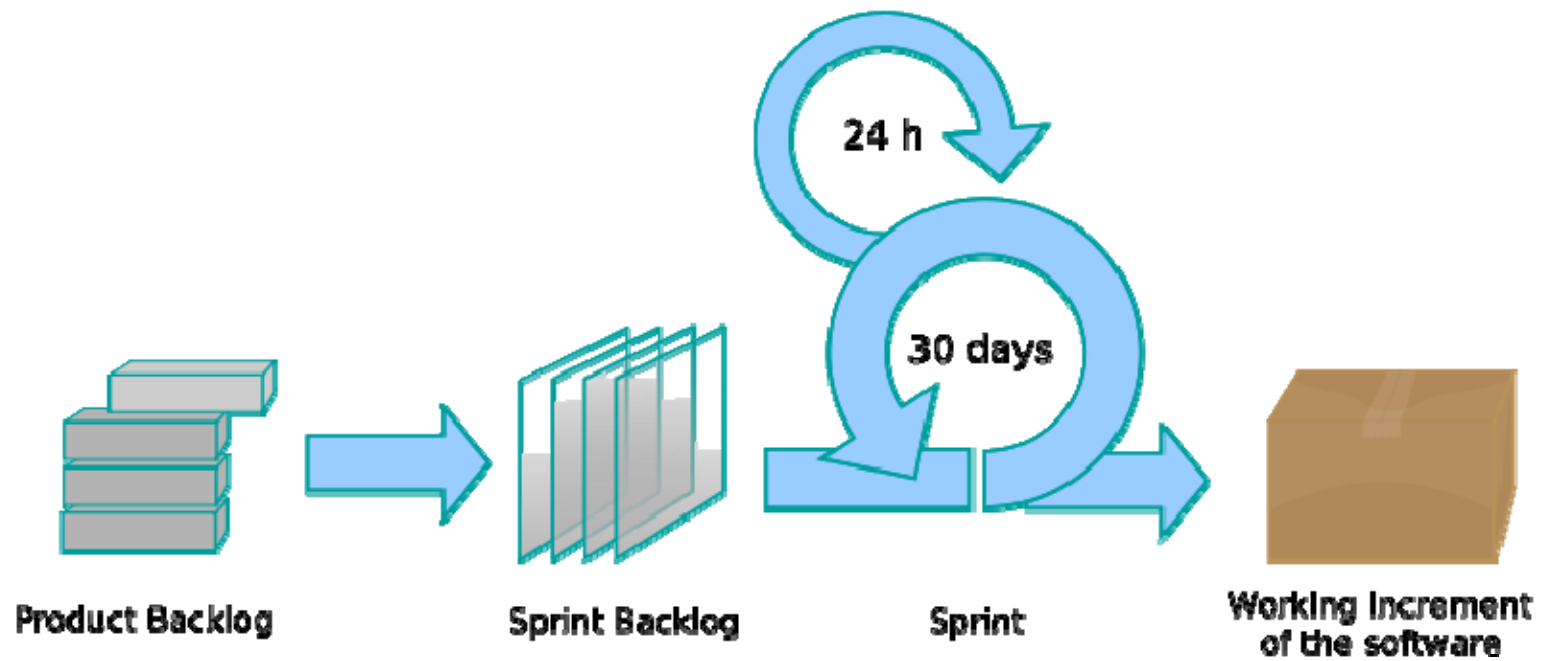
- **Stakeholders (customers, vendors)**

The people that will enable the project, but are only directly involved in the process at sprint reviews.

- **Managers**

People that will set up the environment for the product development organization.

Scrum process flow



Scrum process flow

1. Planning

Product owner and team decide which stories are actually feasible to be moved from the Product backlog to the Sprint backlog.

2. Sprint

The team is left alone to perform the user stories which it has committed itself in the planning meeting. The product owner may attend the “daily scrums” if a granular status update is desired.

3. Review

The team presents its work and verifies what it has done indeed satisfies the utmost desires of the product owner.

User Stories

- A ***user story*** is a software system requirement formulated as one or two sentences in the everyday language of the user.
- It is written by the Product Owner, with the help of the ScrumMaster and Team, if desired and necessary.
- Once completed, it is put in the Product Backlog and prioritized, by the Product Owner, by its relative placement to other user stories.
- Before a user story is to be implemented, appropriate acceptance criteria must be written to ensure proper testing or otherwise determine whether the goals of the user story have been fulfilled.
- Some formalization finally happens when the developer accepts the user story and the acceptance procedure as his work specific order.

User Stories - structure

- **Who (user role)** – is this a customer, employee, system administrator?
- **What (goal)** – What is the specific functionality that is to be achieved or developed?
- **Why (reason)** – Helps the developer to understand the broader scope of the story and eliminate any ambiguities that may arise.
- Putting it all together: ***As a [user role], I want to [goal], so I can [reason].***
- *Exempli gratia:*

“As a registered user, I want to log in, so I can access subscriber content.”

User Stories – I.N.V.E.S.T.

- **Independent** - For some systems, it's near impossible to make each feature completely independent. In other solutions, e.g. web sites, it's easier. But it's an important aspiration. User Stories should be as independent as possible.
- **Negotiable** - User Stories are not a contract. They are not detailed specifications. They are reminders of features for the team to discuss and collaborate to clarify the details near the time of development.
- **Valuable** - User Stories should be valuable to the user (or owner) of the solution. They should be written in user language. They should be features, not tasks.
- **Estimatable** - User Stories need to be possible to estimate. They need to provide enough information to estimate, without being too detailed.
- **Small** - User Stories should be small. Not too small. But not too big.
- **Testable** - User Stories need to be worded in a way that is testable, i.e. not too subjective and to provide clear details of how the User Story will be tested.

Questions?



Presented by:

Bryan Liff

VP, IT Services

bliff@minerva-group.com