



# Scrum



*Making Visions Actionable*

*Pejman Makhfi*

*Certified Scrum Master*

*VP of Solution, Savvion Inc.*

*11/29/2008*





# Challenge

- Development can't estimate and commit on what they do not fully understand
- Business can't freeze scope for a long time and can't sign-off on specs they do not fully understand
- As users see the product, they come up with new ideas





# Solution

Making incremental progress, instead of trying to map out an obstacle course that has never been navigated before!

- Scrum is an iterative incremental process of Agile software development, in contrast to waterfall approach. It is an empirical approach based in process control theory.
- Designed to address changing market demands, evolving requirements, miscommunication and misinterpretation.
- Originally used for software project management, it is now gradually penetrating into other domains.



# Concept

- All activities are time boxed
- Requirements are delivered in form of User Stories (vision driven)  
For example: "A consumer can upload and play videos on the website". Details are added during the course of the project to clarify ambiguity.
- Teams are cross functional including all skills needed to meet the Sprint goal. Ideally with 5-9 members and self-organized.
- Projects are divided into manageable iterations (15 to 30 days cycles) - Sprints
- Work is broken down into smaller activities than the estimating threshold (2 days maximum) - Work Items
- Some "business value" is delivered at the end of each iteration

# Elements

- Three Roles
  - Product Owner
  - ScrumMaster
  - Team
- Three Meetings
  - Sprint Planning
  - Daily Standup Scrum
  - Retrospective
- Three Artifacts
  - Product/Roadmap Backlog
  - Sprint Backlog
  - Burn down Chart

4 hours: Product Owner and team to agree on subset of Product (roadmap) backlog to work on this sprint

4 hours: The team figures out how they are going to deliver the product Increments (Sprint Backlog and task creation)

15 minutes each day. Team talks, managers watch!

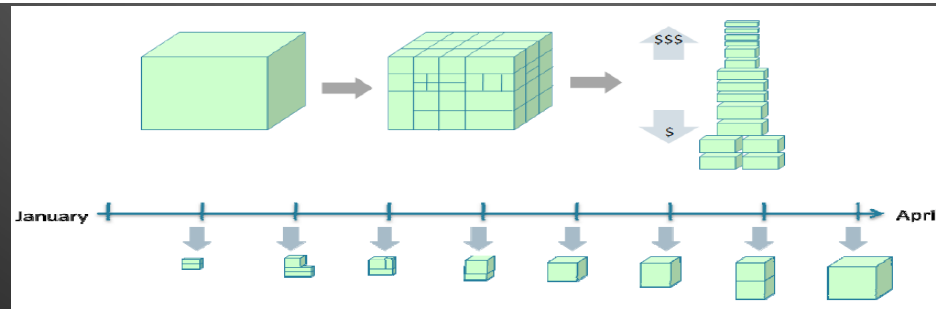
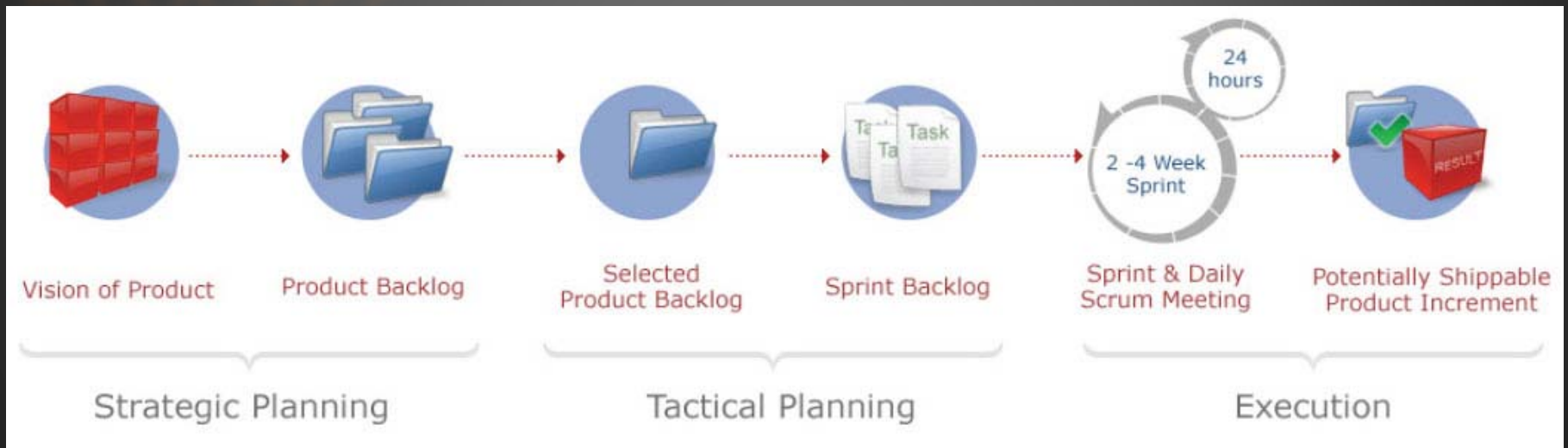
Answering Three questions:

- What did I do since last meeting?
- What has impeded my work?
- What will I do before the next meeting?

Scrum Team to analyze their own process and modify as necessary



# Process



# Process

- Team meets with product owner to discuss Product Backlog Items and assign a relative size value to each.
- Product Owner then prioritizes and discuss with the team the next product increment, as a potentially shippable product.
- Team then breaks down sprint items into a list of small, concrete deliverables (work items) and estimate effort for each.
- At the end of the sprint the team “shows off” what they’ve done to product owner and other Stakeholders, .
- Team meets with ScrumMaster to inspect and adapt on their process.



# Story



- Stories are a way to communicate expectations.
- There are User Stories, Defects and Technical Story Not all requirements for new development represent user-facing features, but do represent significant work that must be done. We call these non-functional requirements "Technical Stories."
- Progress is demonstrated by delivering tested and integrated work that implements a story.
- Scrum teams like to see stories that can be estimated correctly and scheduled in one iteration.





# Story (cont.)

- Each story has a Description, Business Value (BV) as a relative value (e.g. 1-10 range), Story Point (SP) reflecting the relative estimate of effort (e.g. 1-10 range), Definition of Done (DoD) as an instruction on how to test or demo (~ completion criteria)
- Distribute 100 points for BV between stories and you get ROI = BV/SP
- Format:  
As an <Actor/Role>  
I want to <Action/Function>  
so that <Goal/Benefit>

## 1. User stories

### 1.1 Most popular blu-ray discs sold

As a **customer** I want to **see the most popular blu-ray discs sold** so that I **can order one or many of them**

#### 1.1.1 Sort by price

As a **customer** I want to **sort the most sold blu-ray discs by price** so that I **can see the less expensive ones first**

#### 1.1.2 Sort by popularity

As a **customer** I want to **sort the most sold blu-ray discs by popularity** so that I **can see the most popular ones first**



# Epic

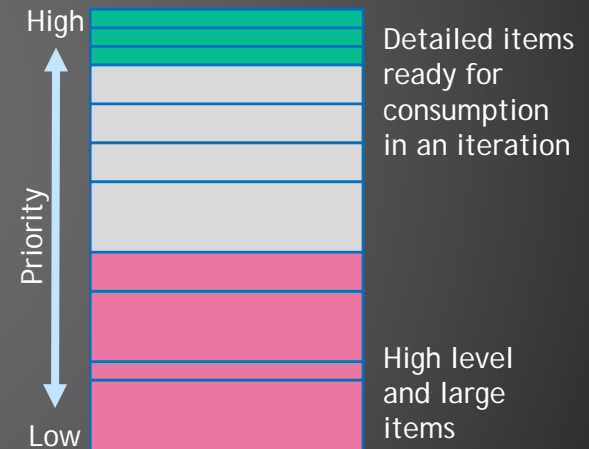


- An Epic is a large story or collection of related stories.
- Some examples of Epics are:
  - We're going to need to train all our users on this new release
  - As a <tourist>, I want to <fly to Catalina for the weekend>
  - As a HR manager I want an internal resume tracking system to simplify my work
  - We want to move our Internet Data Center to another platform
- A "Work In Progress"(WIP) limit of active epics at any given time should be setup



# Product Backlog

- The product backlog has four qualities: It is detailed appropriately, estimated, emergent, and prioritized, making it DEEP
- “Detailed appropriately” means items on top of Backlog should be clearer and have a more detailed specification.
- Items that span over 1 iteration are called Epic
- It may take a few sprints to break down and refine a large Epic until the resulting stories are ready to be pulled into the sprint. (progressive decomposition)



# Sprint Queue

- The Sprint Queue is a subset of Product Backlog Items (Stories and Defects) planned for implementation in a Sprint. The items are ranked in the desired order of implementation to reflect both the urgency (value) of the item, and any potential dependencies that exist between items.
- Ideally each Sprint should have a Theme or Goal. And what features are required from the Product Backlog to support the goal.
- The Sprint turns the Product Backlog for one Sprint into an increment of potentially shippable product.
- Each n Sprints will result to a formal Release



# Work Queue

- Sprint Items are broken down into Work items. They are decompose and ranked in way that takes both customer priority and technical constraints into consideration.
- Work items are tasks that
  - can be done by one person
  - can be done in less that 10 “work hours”
- Detailed task-level estimations are created in the sprint planning meeting. Individual work items and their estimates are captured in the Work Queue.

# Daily Scrum

Daily Scrum refers to a short (15 minutes) daily standup meeting to review progress status and any potential problems preventing team members from accomplishing their goal. ScrumMaster will then facilitate resolution of these impediments outside the context of the Daily Scrum.

Questions to answer by each member:

- What did you do yesterday?  
*I started writing the code to load information into the database.*
- What are you going to do today ?  
*I'm going to handle some of the data issues that cropped up yesterday.*
- Do you have any blockers?  
*I have no blockers.*

For purpose of achieving the same with distributed teams, the queuing solution should be configured to offer automated reports for 1<sup>st</sup> and 2<sup>nd</sup> items. Blockers are shared with the team via email by each member.

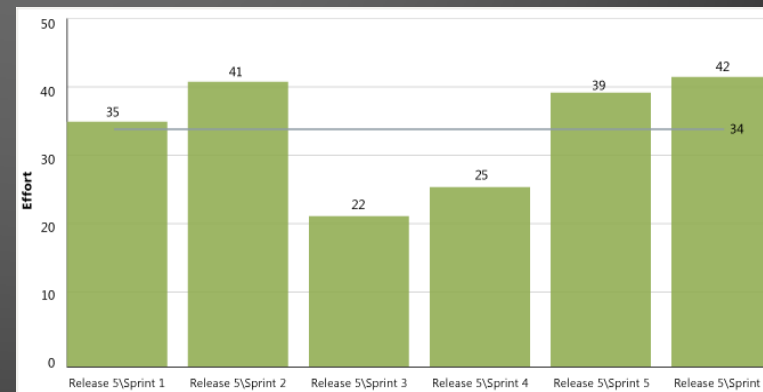
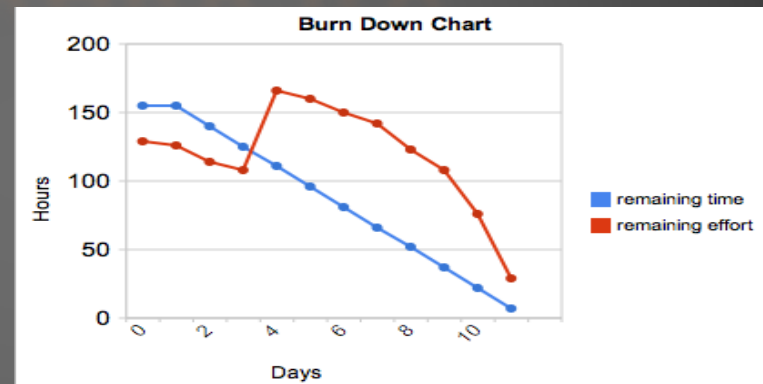
# Sprint Board & Visibility

Story	To Do	In Process	To Verify	Done
<p>As a user, I... 8 points</p>	<p>Code the... 9</p> <p>Code the... 2</p> <p>Test the... 8</p> <p>Test the... 4</p>	<p>Code the... DC 4</p> <p>Test the... SC 8</p>	<p>Test the... SC 6</p>	<p>Code the... D</p> <p>Test the... SC 8</p> <p>Test the... SC</p> <p>Test the... SC</p> <p>Test the... SC 6</p>
<p>As a user, I... 5 points</p>	<p>Code the... 8</p> <p>Code the... 4</p> <p>Test the... 8</p> <p>Code the... 6</p>	<p>Code the... DC 8</p>		<p>Test the... SC</p> <p>Test the... SC</p> <p>Test the... SC 6</p>



# Burn Down & Velocity

- The sprint burn down chart is a publicly displayed chart showing remaining work in the sprint, based on initial WI estimates.
- Velocity is a measurement of how much the team actually gets done in an iteration (and not what was planned)





# Scrum Rules

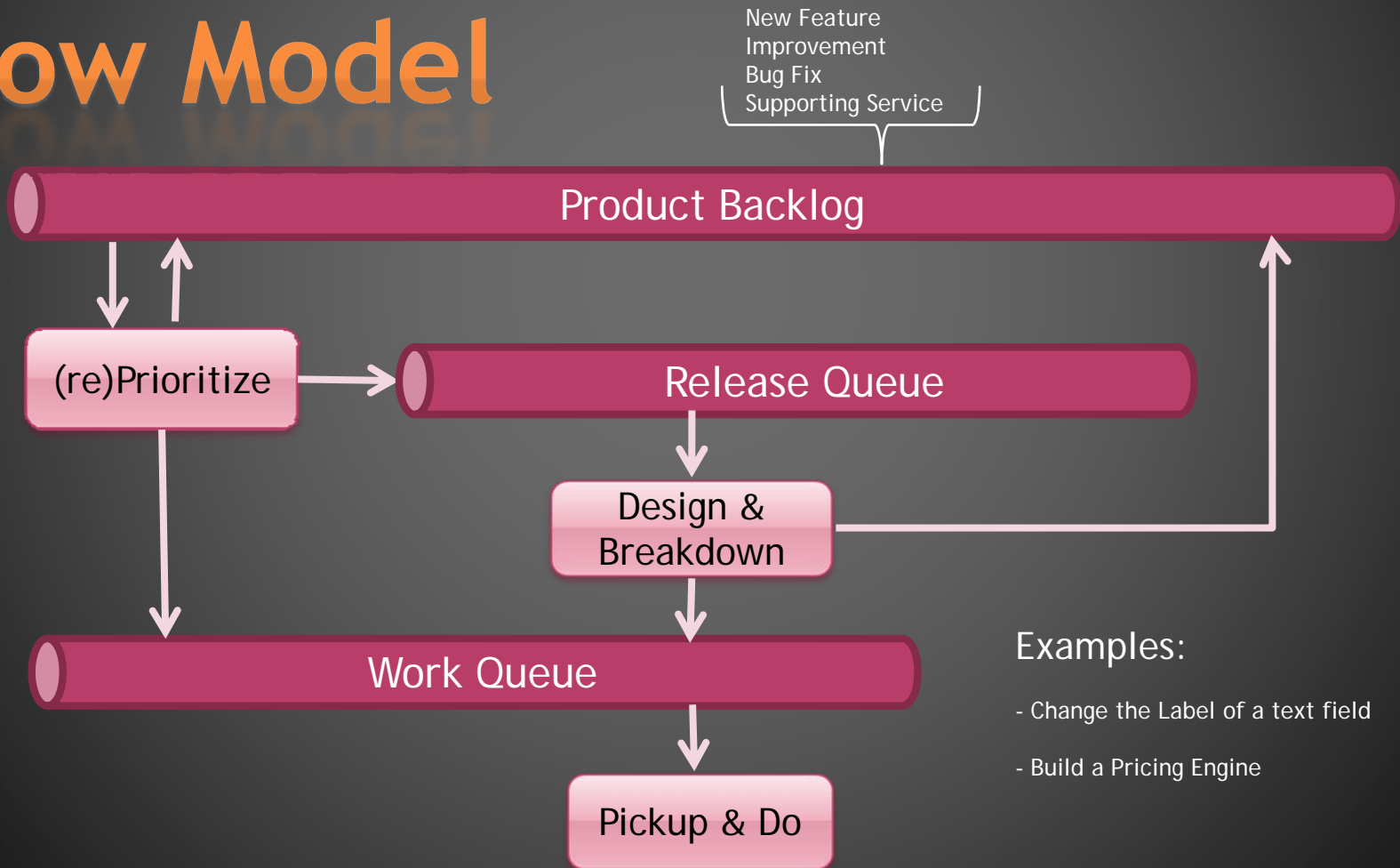
- Each iteration implements the highest priority items
- New Items are prioritized and added to the stack
- Backlog can be reprioritized at any time
- Items can be removed at any time
- Items should have DoD (Definition of Done)
- For some items it might be best to pre-assign
- A percentage of capacity will be allocated for in-flight bugs & unexpected
- There should be a limit to number of WIPs (Work In Progress)
- Estimates might be increased by a Drag Factor depending on the team age and their knowledge of technology and project domain.
- Iterative and incremental is not enough to be agile. It is about adjustment and re-planning

# Implementation Rules

- Effective work capacity to be estimated at 30 hours/week/person
- Every activity greater than 1 hour has to be captured in the system
- Maximum effort per WI can be 10 hours, otherwise break down
- Estimates are captured at WI level and rolled up to each Story
- For integration and Items that need testing beyond unit test there will be explicit WIs
- The WorkQueue has to be emptied before the Sprint end. That is a Team responsibility and not of any individual member, regardless of task type.



# Flow Model



## Examples:

- Change the Label of a text field
- Build a Pricing Engine





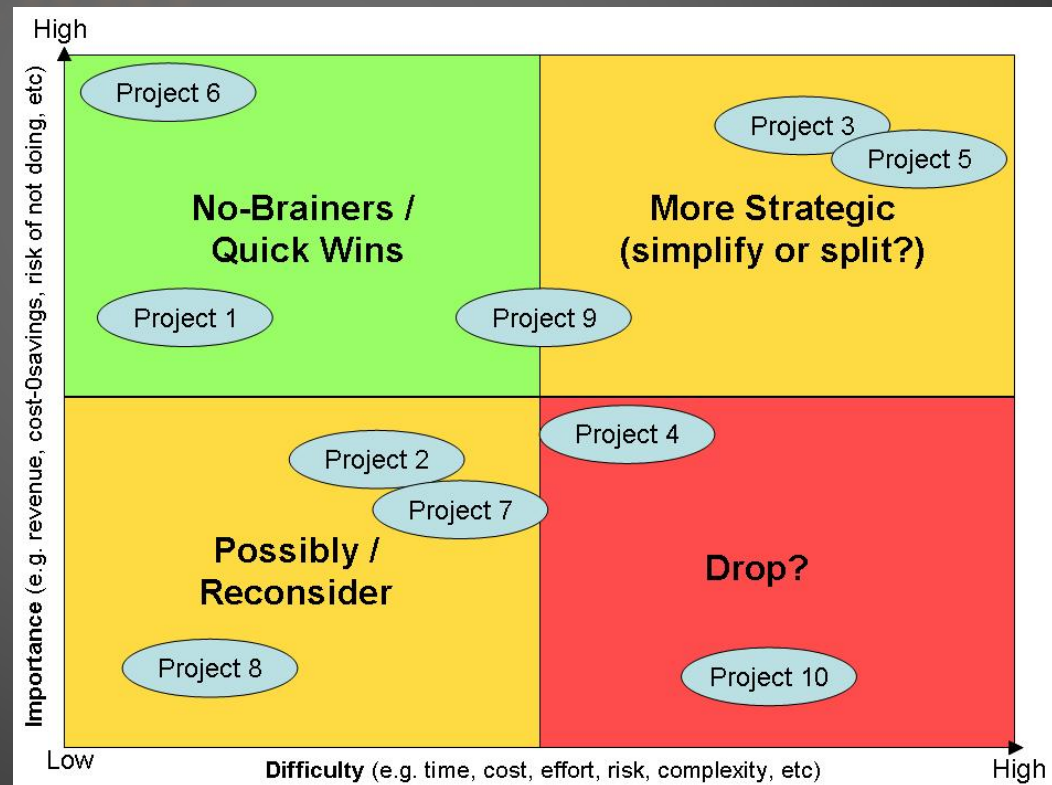
# *Appendix*



# Prioritization

Prioritization is difficult with complex and conflicting objectives or when benefits are intangible. In that case use the \$35 rule!

Distribute \$35 between features, according to their business values



# Analysis & Design

*At design level requirements should be translated to following aspects:*

- Usage Model (Use cases and application logic)
- Data Model (entities and relationships)
- Interaction Models (UI/Human and API/System interfaces)



# Continuous Integration & Test

Without automated build and tests it is very difficult to work in short development cycles. Tools to consider:

- jUnit / jsUnit
- Selenium systems test
- Cucumber
- BrowserMob load tests
- ANT auto build & deployment

