РАЗВОЈ СОФТВЕРА 2

Агилне методологије развоја софтвера

Scrum

АГИЛНЕ МЕТОДОЛОГИЈЕ

SOFTWARE DEVELOPMENT LIFECYCLE

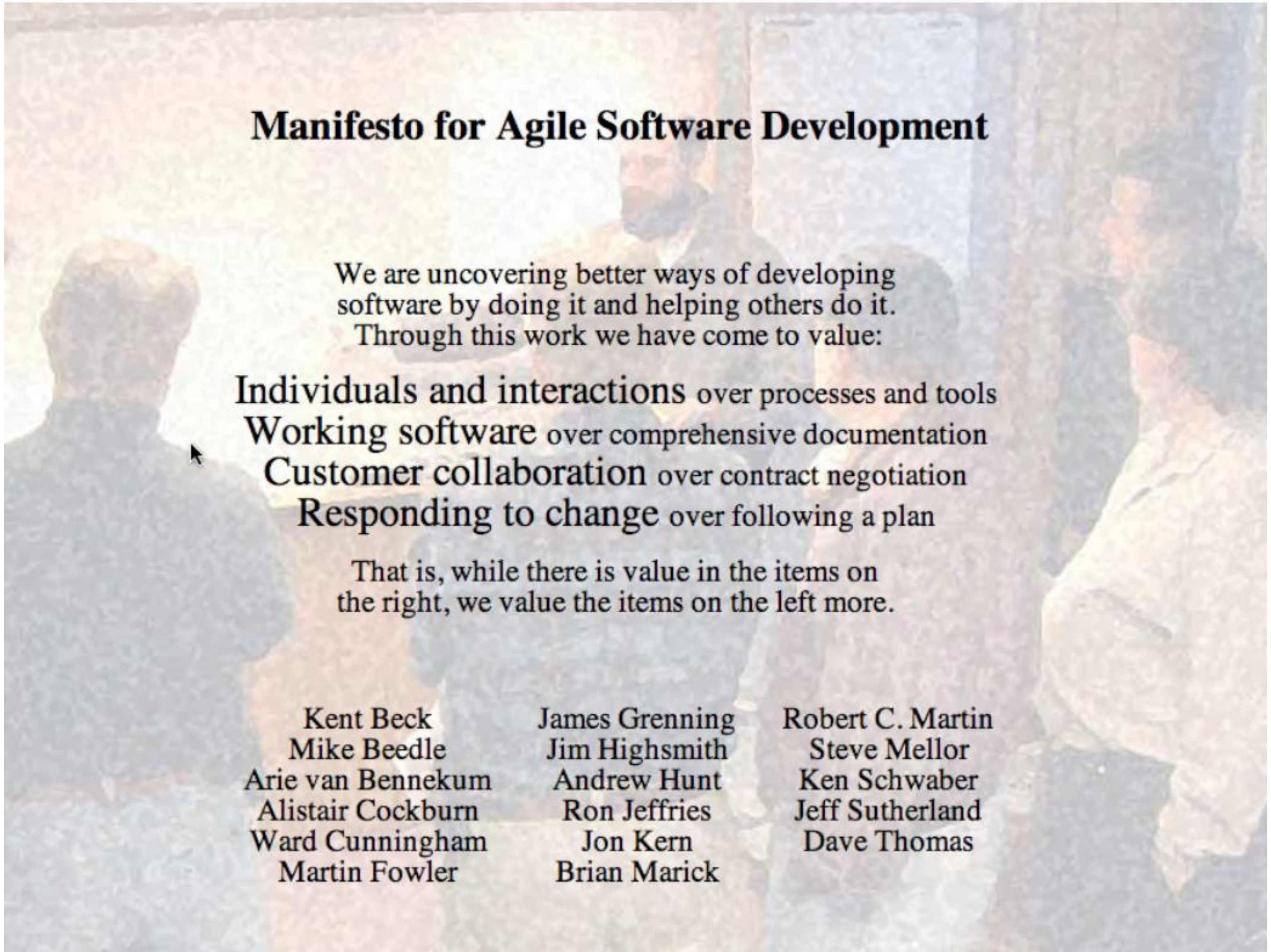
- Waterfall Model
- Iterative Model
  - Spiral Model
  - Agile Methodology
    - Extreme Programming (XP)
    - Test Driven Development (TDD)
    - Scrum
    - Lean Software Development
    - …………………

WHAT IS AGILE METHODOLOGY?

- Project management with frequent inspection and adaptation
- Encourages teamwork, self-organization and accountability
- Testers need understanding of agile as a whole
- Also need to learn how to adapt their current approach to work within agile efforts

AGILE MANIFESTO

http://agilemanifesto.org/



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

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.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

AGILE MANIFESTO

A statement of values:

- Individuals and interactions *over* *processes and tools*
- Working software *over* *comprehensive documentation*
- Customer collaboration *over* *contract negotiation*
- Responding to change *over* *following a plan*

PRINCIPLES BEHIND THE AGILE MANIFESTO

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.

PRINCIPLES BEHIND THE AGILE MANIFESTO

- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

PRINCIPLES BEHIND THE AGILE MANIFESTO

- Continuous attention to technical excellence and good design enhances agility.
- Simplicity -- the art of maximizing the amount of work not done -- is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

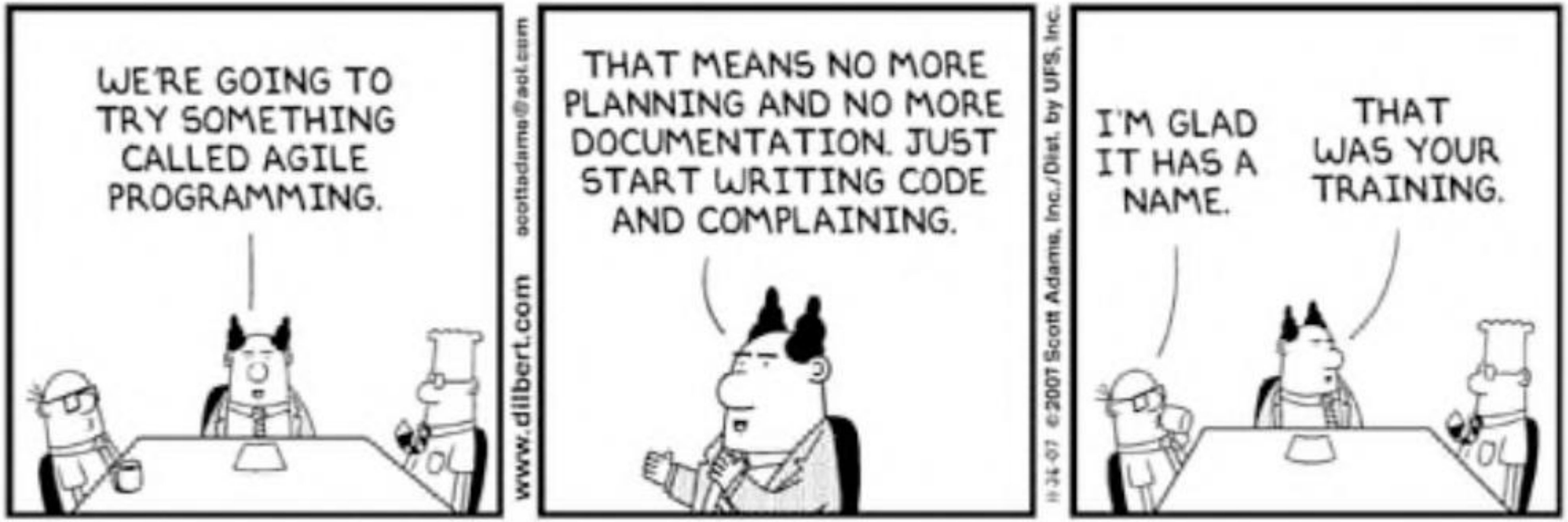
BENEFITS OF AGILE

- Ability to manage changing priorities.
- Improved project visibility.
- Faster time to market.
- Enhanced S/w quality.
- Reduce risk.
- Low cost.
- Customer collaboration.

SOME DISADVANTAGES

- Lack of importance on Documentation.
- Time limit.
- Need better Team work.
- More management effort required for monitoring and controlling activities.

JOKE ABOUT AGILE ☺


WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
www.dilbert.com scottadams@aol.com
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
11-24-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.

© Scott Adams, Inc./Dist. by UFS, Inc.

AGILE PROCESSES

- XP and TDD (code and test centric practices)
- RUP-lite and dX
- Scrum (management practices)
- FDD (Feature Driven Development)
- DSDM (Dynamic System Development Method)

ЕКСТРЕМНО ПРОГРАМИРАЊЕ И РАЗВОЈ УПРАВЉАН ТЕСТОВИМА

EXTREME PROGRAMMING

- XP is perhaps the most widely recognized agile development method
  - Based on taking successful development practices
- Covers both code and business perspectives
  - Agile methods considered to be code-centric, but most do not offer code-level practices

XP Values

- At the heart of XP are five values...

  Communication
  Simplicity
  Feedback
  Courage
  Respect

XP PRIMARY PRACTICES

- ➔ XP is based on thirteen primary practices
  - ➔ Each individually makes sense on its own...
  - ➔ But they work together to reinforce one another
- ➔ XP is therefore a highly disciplined approach

- Sit together
- Whole team
- Informative workspace
- Energized work
- Pair programming
- Stories
- Weekly cycle
- Quarterly cycle
- Slack
- Ten-minute build
- Continuous integration
- Test-Driven Programming
- Incremental Design

XP COROLLARY PRACTICES

- The eleven corollary XP practices supplement the primary practices
  - Some are more speculative than the primary practices
  - The primary practices should be in place before the corollary ones

- **Real customer involvement**
- **Incremental deployment**
- **Team continuity**
- **Shrinking teams**
- **Root-cause analysis**
- **Shared code**
- **Code and tests**
- **Single code base**
- **Daily deployment**
- **Negotiated scope contract**
- **Pay-per-use**

XP FINAL REMARKS

➔ Adoption of XP is based on practices not on aping surface phenomena

E.g. avoiding comments or documentation does not mean that you are doing XP!

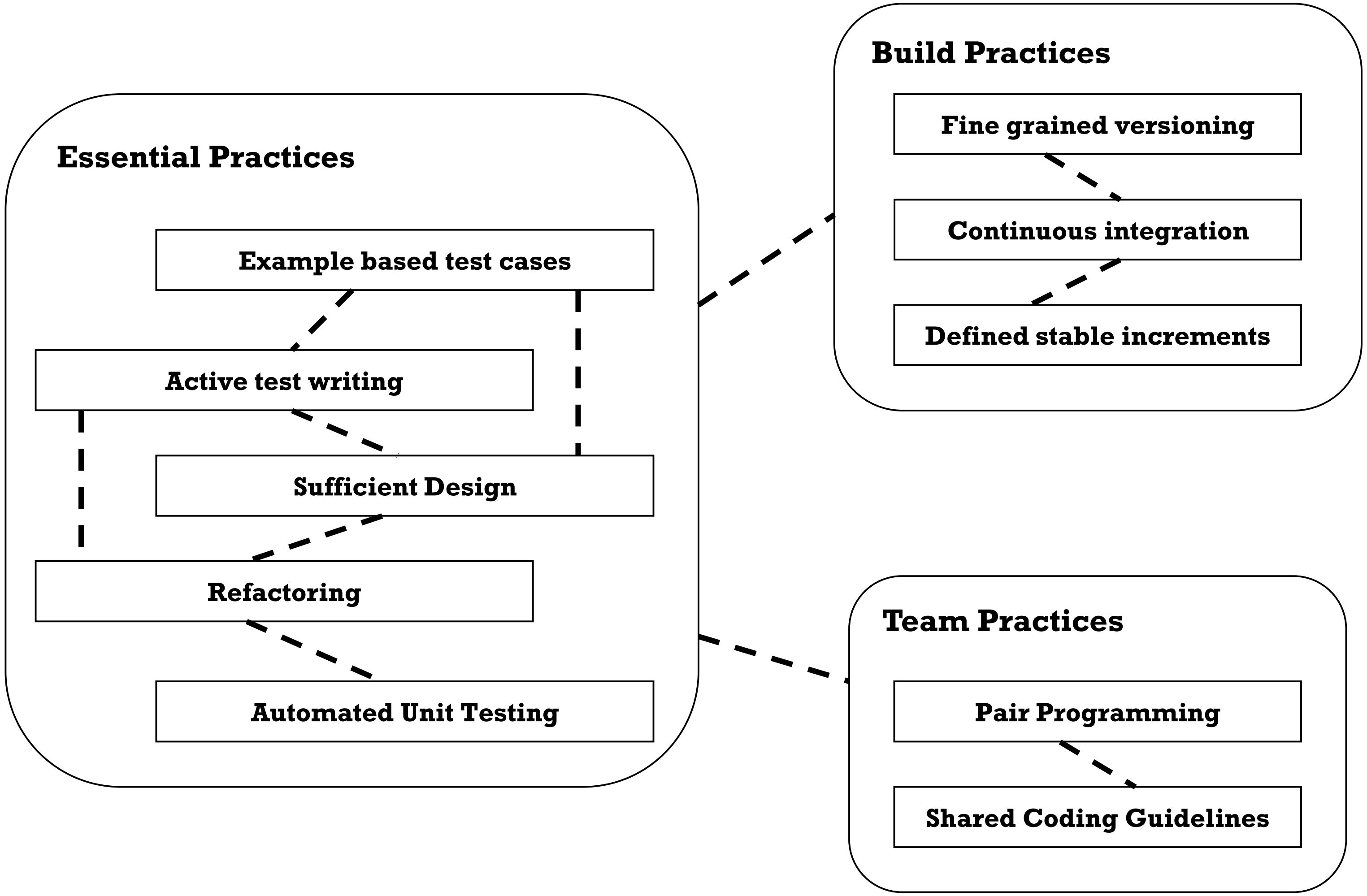
➔ Core practices must be in place

E.g. cannot be XP if there is no refactoring, testing, pairing or continuous integration

TEST DRIVEN DEVELOPMENT

- ➔ TDD has emerged from the many practices that form the core of Extreme Programming
  - ➔ The code-centric practices in the micro-process rather than driving the macro-process
- ➔ TDD brings testing to the fore of development rather than the after

AGILE CODE AND TEST CENTRIC PRACTICES



Essential Practices

Example based test cases

Active test writing

Sufficient Design

Refactoring

Automated Unit Testing

Build Practices

Fine grained versioning

Continuous integration

Defined stable increments

Team Practices

Pair Programming

Shared Coding Guidelines

Scrum

WHAT IS Scrum?

- Scrum is an Agile Process
  - Focus on delivering the highest business value in the shortest time
  - Repeatedly inspect actual working software (every two weeks to one month)
- The business sets the priorities
  - Teams self-manage to determine the best way to deliver the highest priority features
- Strive towards multiple Releases
  - Every two weeks to a month evaluate working software
  - Development Team and Domain Experts
  - Decide to release it as is or continue to another iteration

HISTORY OF Scrum…

- **1995** Analysis of common software development processes
  - Design of a new method: Scrum by Jeff Sutherland & Ken Schwaber
  - Enhancement of Scrum by Mike Beedle & combination of Scrum with Extreme Programming
- **1996:** Intro of Scrum at OOPSLA conference
- **2001:** "Agile Software Development with Scrum" by Ken Schwaber & Mike Beedle

→Successful appliance of Scrum in over 50 companies

Founders are members in the Agile Alliance

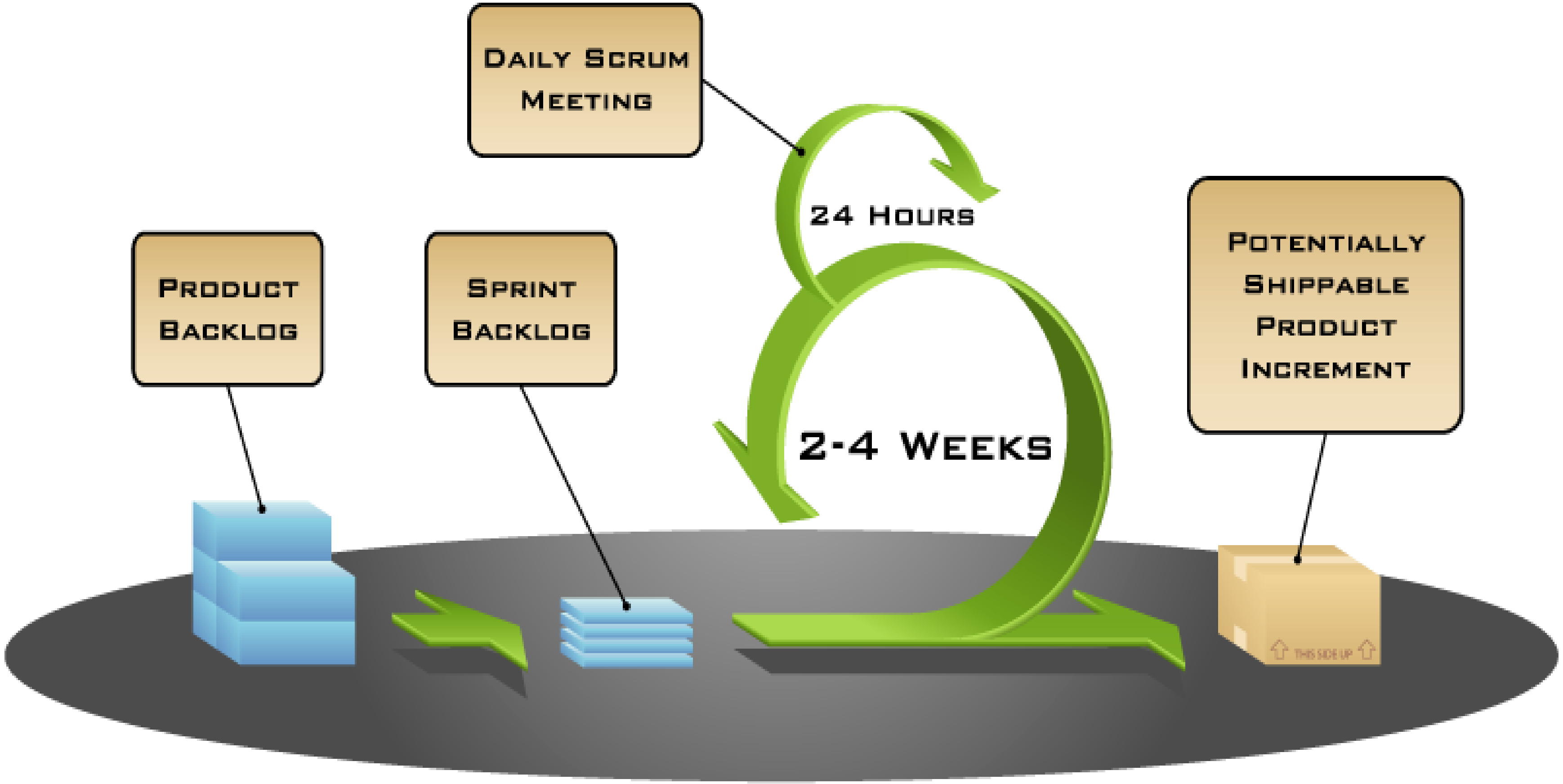
CHARACTERISTICS OF SCRUM…

- Self-organizing teams
- Product progresses in a series of month-long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- One of the “agile processes”

WHY SCRUM IS POWERFUL..?

- Focus is on team’s work and team’s work only
- Daily communication of status occurs
- Enables low-overhead empirical management
- Makes impediments visible
- Someone is willing to make decisions and
- Remove impediments real-time

HOW DOES IT WORKS?



Copyright © 2005, Mountain Goat Software

COMPONENTS OF SCRUM…

- A process for incrementally building software in complex environments
- Backlog – all outstanding work for a product area
- Sprints – 30-day increments of work that produce

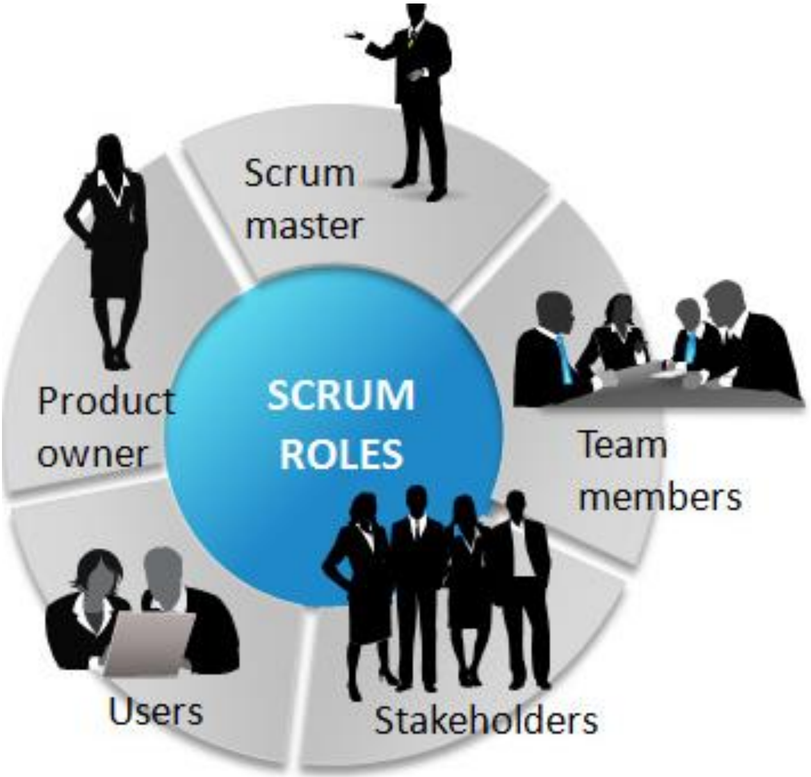
  a deliverable
- Scrums – daily status check meetings

SCRUM MEMBERS…

- Typically 5-10 people
- Cross-functional
  - QA, Programmers, UI Designers, Testers, DB, etc.
- Members should be full-time
  - May be exceptions (e.g., System Admin, etc.)
- Teams are self-organizing
  - What to do if a team self-organizes someone off the team??
  - Ideally, no titles but rarely a possibility
- Membership can change only between sprints

SCRUM ROLES

- The Product Owner (product owner will be stakeholder also)
- The Scrum Master
- The Development Team
- Stakeholders



PRODUCT OWNER

✓The Product Owner is responsible for delivering product value.

✓The Product Owner prioritizes the list of items on the Product Backlog.

✓Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next.

✓Create and maintain the Release Burndown Chart.

✓Clearly communicate the business case to the Team and Stakeholders.

✓Report progress to the Stakeholders regularly.

SCRUM MASTER

✓ The Scrum Master is responsible for ensuring Scrum is understood and enacted.

✓Facilitate Sprint Planning, Daily Scrums, Sprint Reviews and Retrospective Meetings.

✓Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value.

THE DEVELOPMENT TEAM

- ✓The Team is ultimately responsible for committing to a Sprint goal and promising to deliver it within the time boxed Sprint.
- ✓The Team is self-managing and self-organized.
- ✓Team commits to the Sprint, they must ensure that they can deliver on what they promised.
- ✓Help create and maintain the Sprint Backlog, Sprint Burndown Chart and Task Board.
- ✓Demonstrate the product at the end of each Sprint — during the Sprint Review.

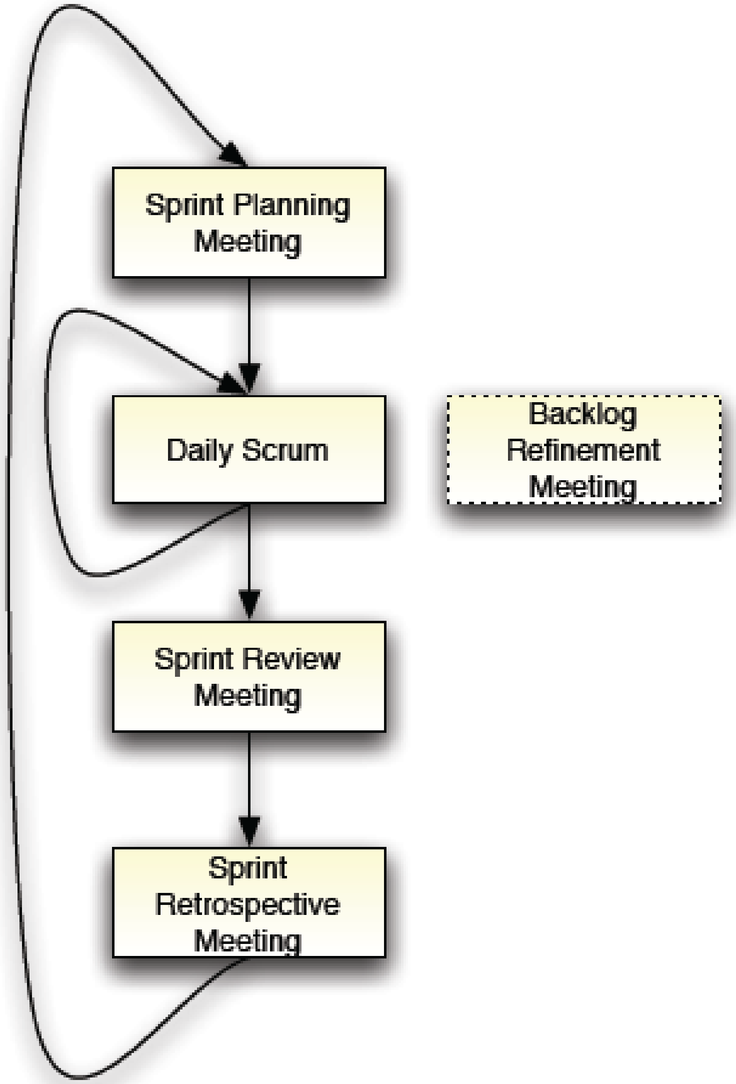
STAKEHOLDER

- ✓The Stakeholder is anyone who has an interest or stake in the project. This can be the direct managers of the Team members, the persons providing funding for the project as well.
- ✓Stakeholders are responsible for communicating their needs, and providing feedback on the product.
- ✓Provide direct feedback to the Team during Sprint Reviews.

SCRUM EVENTS

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

PROCESS VIEWPOINT



SPRINT PLANNING

The work to be performed in the Sprint is planned at the Sprint Planning.

Sprint Planning meetings are time-boxed events that determine which product backlog items will be delivered and how the work will be achieved.

Topic Discussed In Sprint Planning Meeting :

- ✓What can be done in this Sprint.
- ✓How will the chosen work get done?
- ✓Sprint goal.

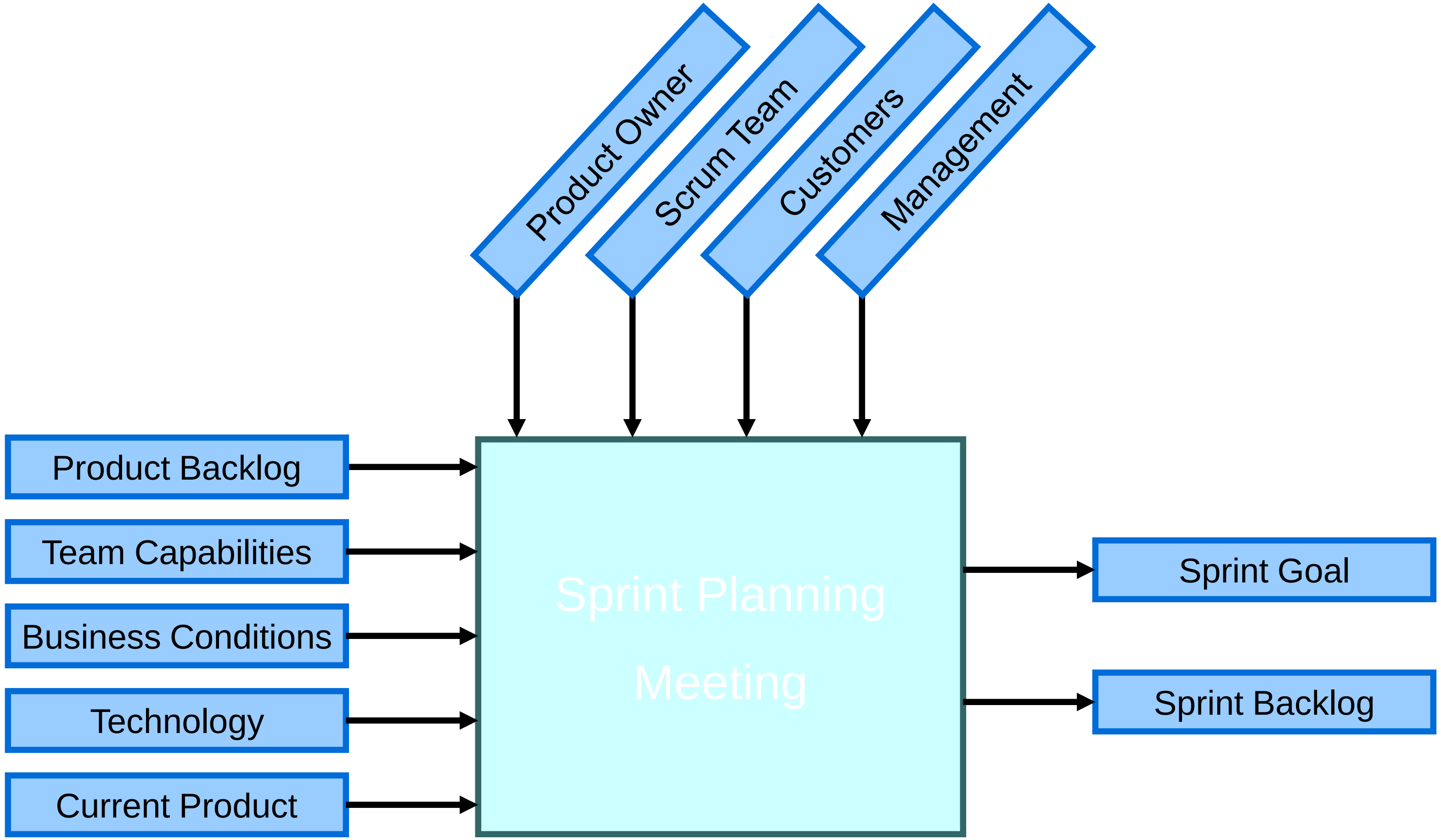
SCRUM SPRINT RULES…

- Use small interdisciplinary teams
- Build clean interface software
- Intelligent management required
- Solid systems architecture and framework upfront
- Prototype all new tools and technology
- Develop infrastructure first
- Each Sprint results in an executable
- Develop, document, and test in parallel

PRODUCTS SPRINT PLANNING MEETING…

- 1st Part:
  - Creating Product Backlog
  - Determining the Sprint Goal
  - Participants: Product Owner, Scrum Master, Scrum Team
- 2nd Part:
  - Participants: Scrum Master, Scrum Team
  - Creating Sprint Backlog

Sprint Planning Meeting….



DAILY SCRUM

The Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours. The Daily Scrum is held at the same time and place each day to reduce complexity. The Development Team uses the Daily Scrum to inspect progress toward the Sprint Goal

Topic Discussed In Daily Scrum :

- ✓What did I do yesterday.
- ✓What will I do today.
- ✓Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

DAILY SCRUM…

- Is NOT a problem solving session
- Is NOT a way to collect information about WHO is behind the schedule
- Is a meeting in which team members make commitments to each other and to the Scrum Master
- Is a good way for a Scrum Master to track the progress of the Team

DAILY SCRUM QUESTIONS…

- What did you do since the last Scrum?
- What got in your way?
- What are you going to do before the next Scrum?

SPRINT REVIEW

During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint. The Sprint Review is the demonstration event for the team to present the work completed during the sprint. The stakeholders or clients give feedback to ensure that the delivered increment met the business need.

Topics Discussed In Sprint Review Meeting :

- ✓The Product Owner explains what Product Backlog items have been "Done" and what has not been "Done".
- ✓The Development Team demonstrates the work that it has "Done".
- ✓The Product Owner discusses the Product Backlog. The Product Backlog may also be adjusted overall to meet new opportunities.

SPRINT REVIEW MEETING…

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
  - 2-hour prep time rule

- Participants
  - Customers
  - Management
  - Product Owner
  - Other engineers

SPRINT RETROSPECTIVE

The Retrospective, or Retro, is the final team meeting in the Sprint to determine what went well, what went wrong, and how the team can improve in the next Sprint. The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning.

Topics Discussed In Sprint Retrospective Meeting :

- ✓What went well & what went wrong.
- ✓Create a plan for implementing improvements for remaining sprints.

SCRUM TERMINOLOGY

- Sprint
- User Story
- Product Backlog
- Sprint Backlog
- Definition Of Done
- Burn down Chart

SPRINT

In the Scrum method of Agile software development, work is confined to a regular, repeatable work cycle, known as a sprint or iteration. Scrum sprints used to be 30 days long, but today many teams prefer shorter sprints, such as one-week or two-week sprints. A new Sprint starts immediately after the conclusion of the previous Sprint.

During the Sprint:

- No changes are made that would affect the Sprint Goal.
- Sprint can’t be delay.
- Quality goals do not decrease.
- Only the Product Owner has the authority to cancel the Sprint.

USER STORY (REQUIREMENT)

A user story is a tool used in Agile software development to capture a description of a software feature from an end-user perspective. An Agile user story is meant to be short.
The user stories should be written by the business in the language of the customer so that it is clear to both the business and the development team what the customer wants and why he wants it. A user story is a very high-level definition of a requirements. Each story is having story point.

There will be 2 things in each User Story :

1. Acceptance Criteria
2. Business Rules.

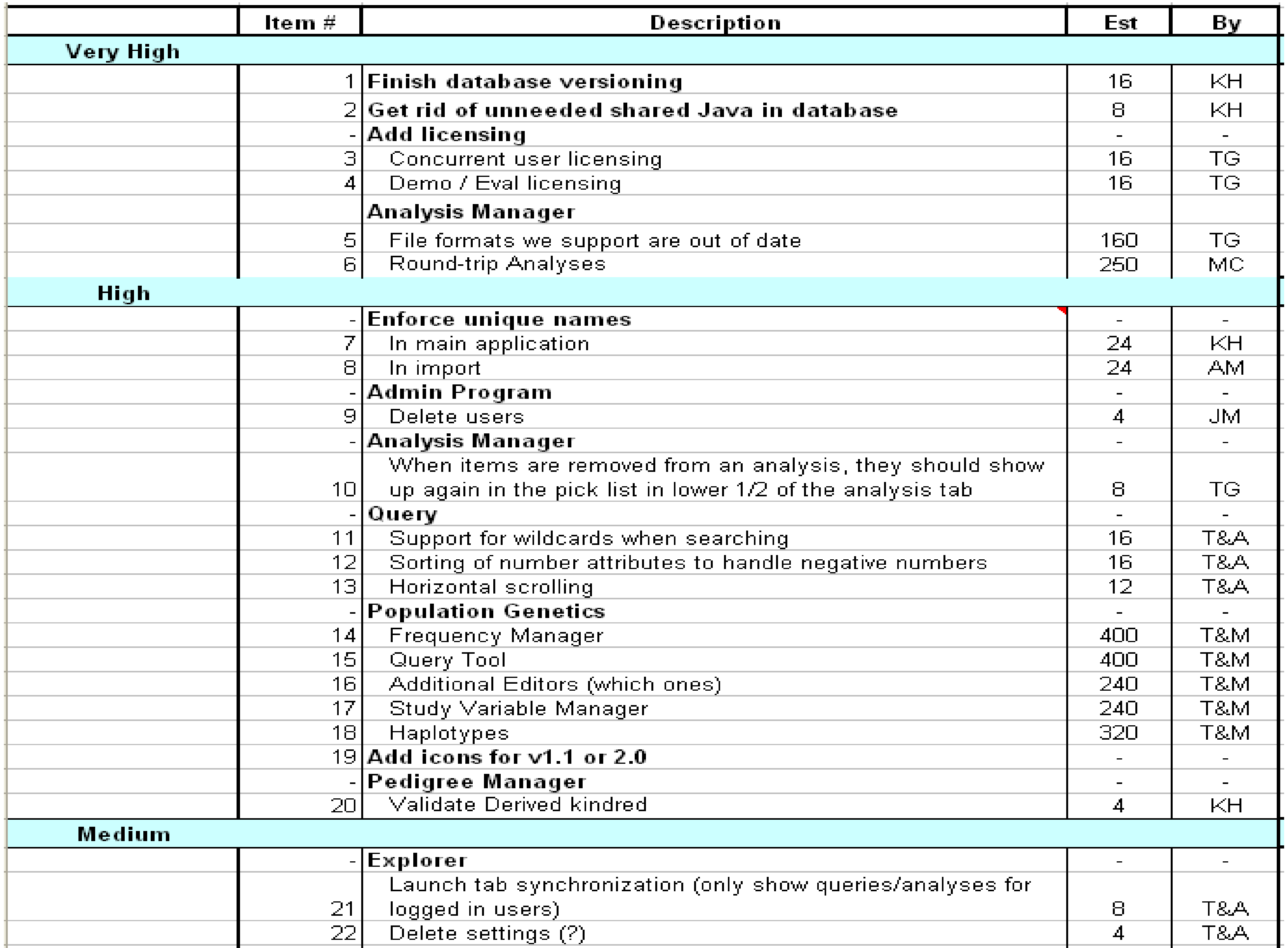
PRODUCT BACKLOG

- The Product Backlog is an ordered list of everything that might be needed in the product.
- The Product Owner is responsible for the Product Backlog including its content, availability, and ordering (Priority basis).
- A Product Backlog is never complete, As long as a product exists, its Product Backlog also exists.
- The Product Backlog lists all features, functions, requirements and enhancements that constitute the changes to be made to the product in future releases.
- Requirements never stop changing, so a Product Backlog is a living artifact.

PRODUCT BACKLOG…

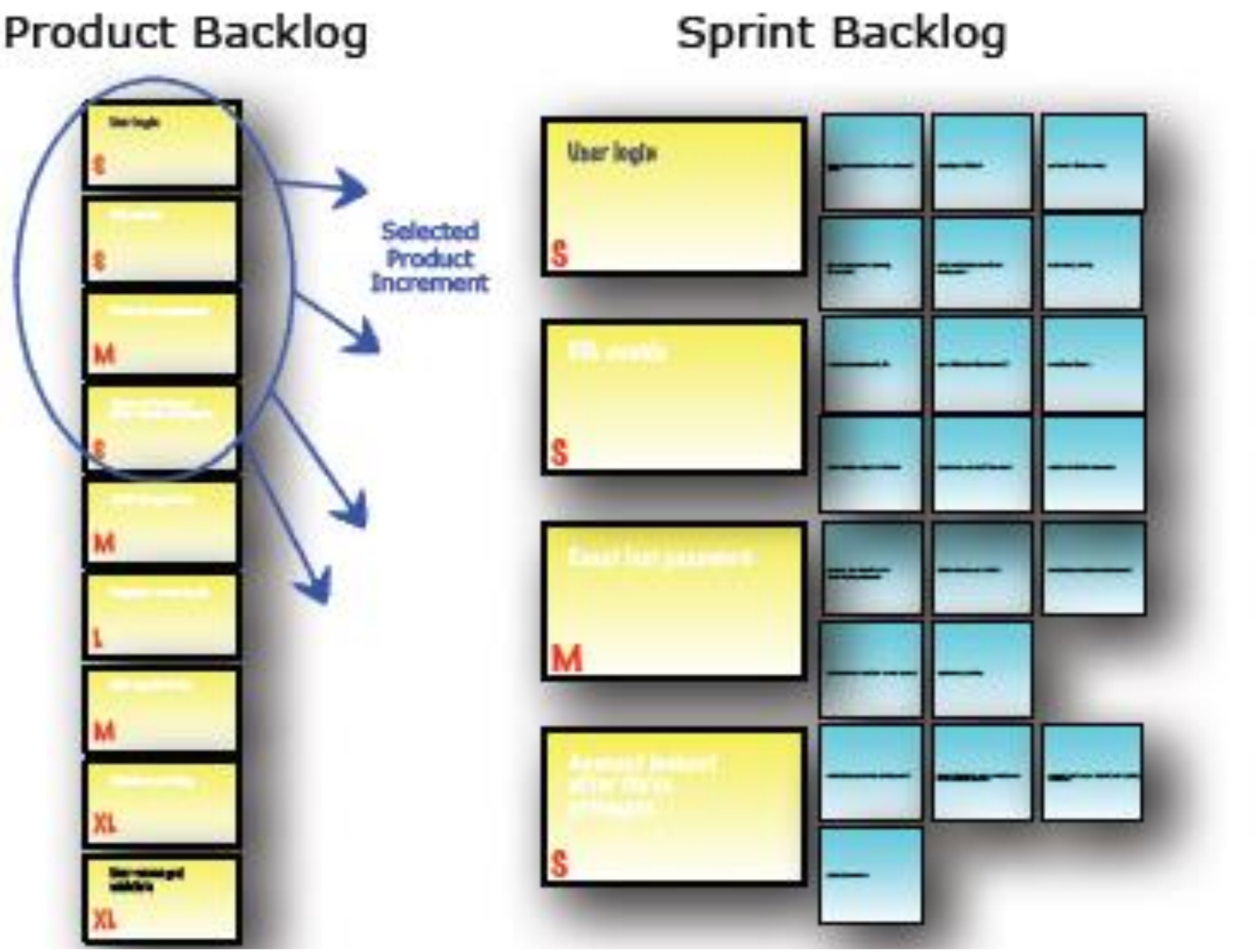
- A list of all desired work on the project
- List is prioritized by the Product Owner
  - Typically a Product Manager, Marketing, Internal Customer, etc.
- Requirements for a system, expressed as a prioritized list of Backlog Items
- Is managed and owned by a Product Owner
- Usually created during Sprint Planning Meeting

SAMPLE PRODUCT BACKLOG…



	Item #	Description	Est	By
Very High				
	1	**Finish database versioning**	16	KH
	2	**Get rid of unneeded shared Java in database**	8	KH
	-	**Add licensing**	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
	-	**Enforce unique names**	-	-
	7	In main application	24	KH
	8	In import	24	AM
	-	**Admin Program**	-	-
	9	Delete users	4	JM
	-	**Analysis Manager**	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
	-	**Query**	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
	-	**Population Genetics**	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	**Add icons for v1.1 or 2.0**	-	-
	-	**Pedigree Manager**	-	-
	20	Validate Derived kindred	4	KH
Medium				
	-	**Explorer**	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

VARIOUS BACKLOGS



SPRINT BACKLOG

- A sprint backlog is the specific list of items taken from the product backlog which are to be completed in a sprint.
- The Sprint Backlog makes visible all of the work that the Development Team identifies as necessary to meet the Sprint Goal.
- The Development Team modifies the Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint.
- As new work is required, the Development Team adds it to the Sprint Backlog.
- As work is performed or completed, the estimated remaining work is updated.
- The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint.

SPRINT BACKLOG…

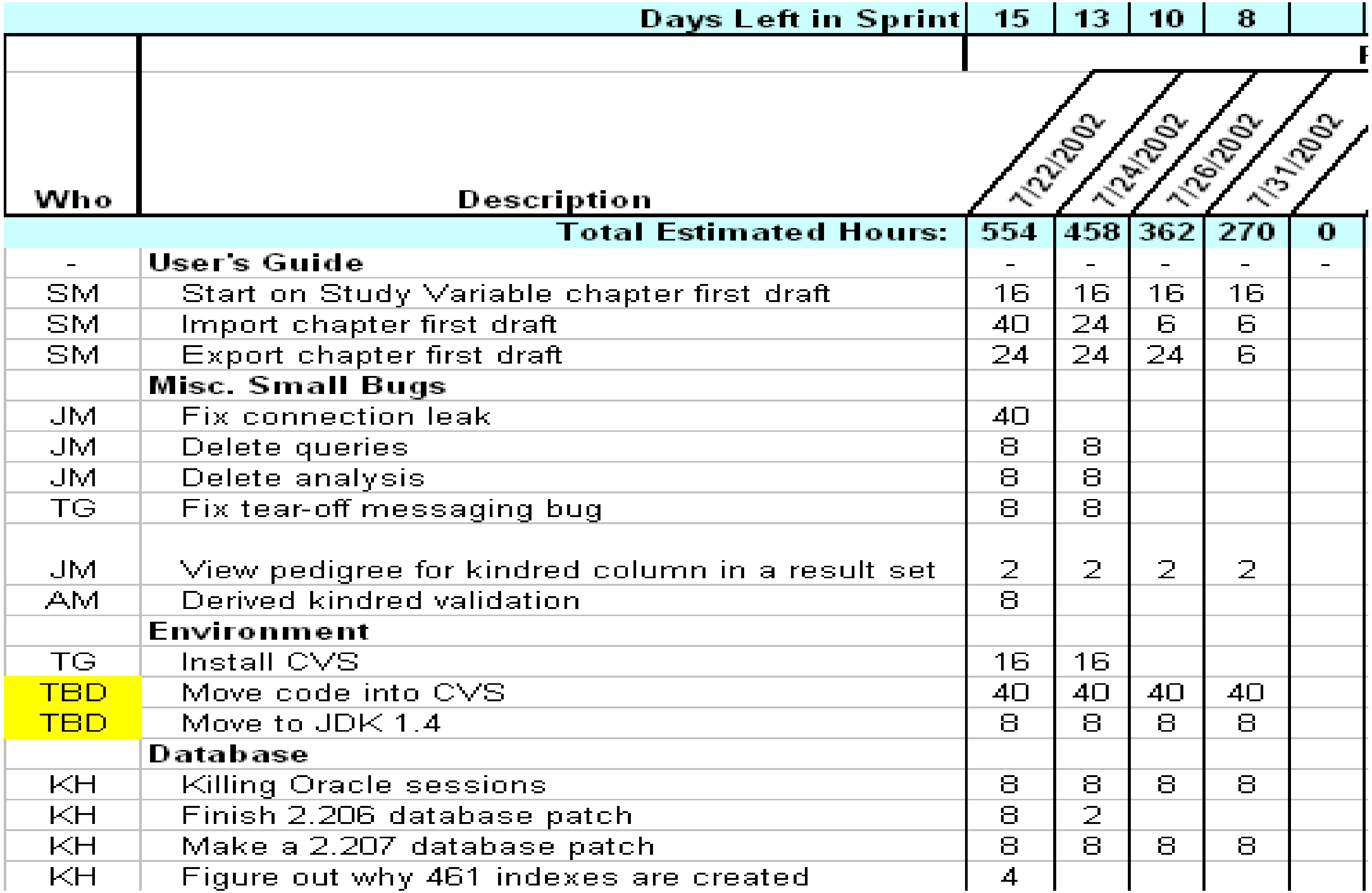
- A subset of Product Backlog Items, which define the work for a Sprint
- Is created ONLY by Team members
- Each Item has it's own status
- Should be updated every day
- No more than 300 tasks in the list
- If a task requires more than 16 hours, it should be broken down
- Team can add or subtract items from the list
- Product Owner is not allowed to

SPRINT BACKLOG DURING THE SPRINT…

- Changes
  - Team adds new tasks whenever they need to in order to meet the Sprint Goal
  - Team can remove unnecessary tasks
  - But: Sprint Backlog can only be updated by the team

- Estimates are updated whenever there's new information

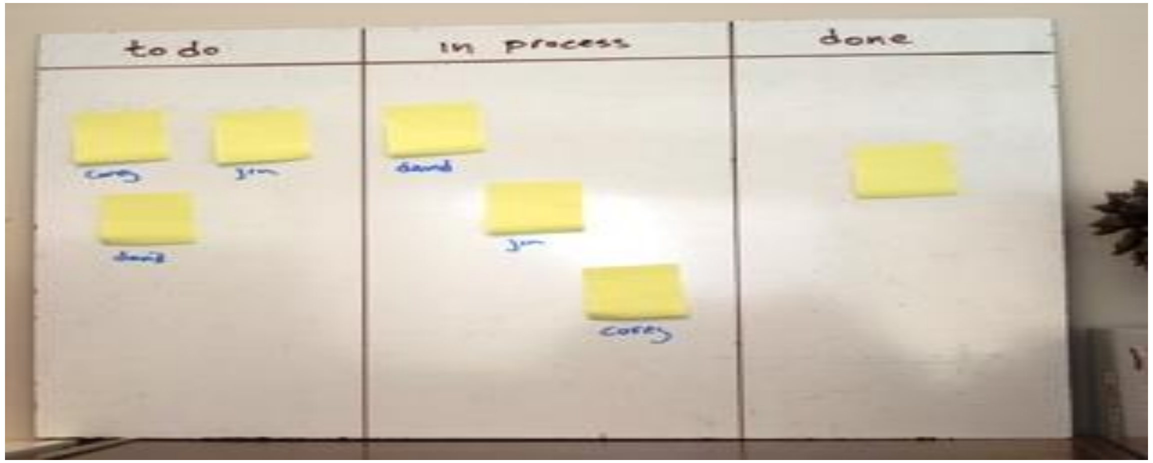
SPRINT BACKLOG…



	Days Left in Sprint	15	13	10	8	
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002	
	Total Estimated Hours:	554	458	362	270	0
-	**User's Guide**	-	-	-	-	-
SM	Start on Study Variable chapter first draft	16	16	16	16	
SM	Import chapter first draft	40	24	6	6	
SM	Export chapter first draft	24	24	24	6	
	Misc. Small Bugs					
JM	Fix connection leak	40				
JM	Delete queries	8	8			
JM	Delete analysis	8	8			
TG	Fix tear-off messaging bug	8	8			
JM	View pedigree for kindred column in a result set	2	2	2	2	
AM	Derived kindred validation	8				
	Environment					
TG	Install CVS	16	16			
TBD	Move code into CVS	40	40	40	40	
TBD	Move to JDK 1.4	8	8	8	8	
	Database					
KH	Killing Oracle sessions	8	8	8	8	
KH	Finish 2.206 database patch	8	2			
KH	Make a 2.207 database patch	8	8	8	8	
KH	Figure out why 461 indexes are created	4				

DEFINITION OF DONE (DOD)

- Definition of done is crucial to a highly functioning Scrum team.
- DoD will ensure that you are delivering features that are truly done, not only in terms of functionality but in terms of quality as well.
- If there are multiple Scrum Teams working on the system or product release, the development teams on all of the Scrum Teams must mutually define the definition of "Done."

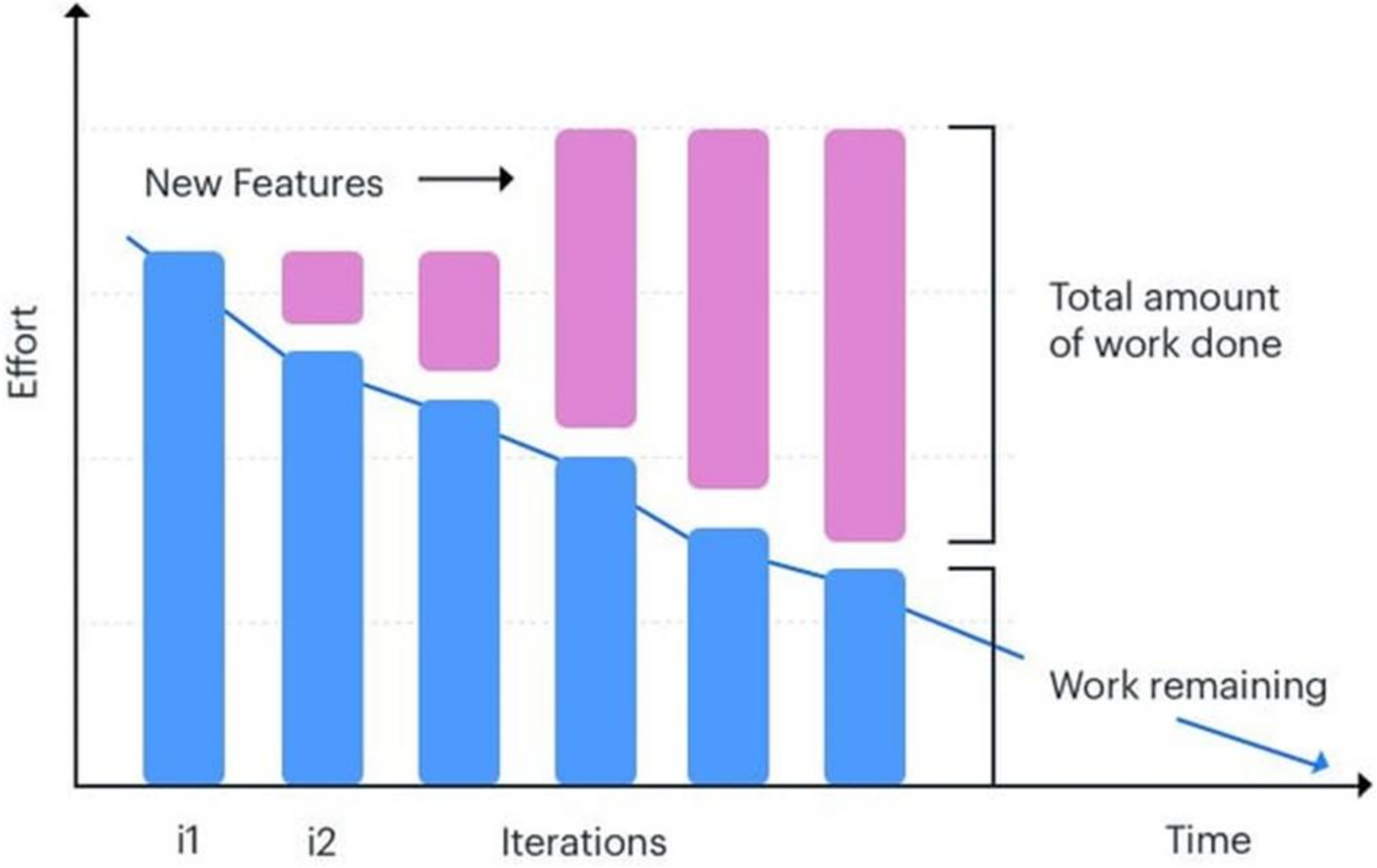


BURN-DOWN CHART

Burn-downs charts are among the most common *sprint tracking mechanisms* used by Agile practitioners.
The first step is to have a task breakdown in place. This is generally done during the sprint planning meeting. Each task should have associated hours (Complete Sprint), which the team decides on during the planning meeting. Once the task breakdown is in place, the ideal burn-down chart is plotted. The ideal reflects progress assuming that all tasks will be completed within the sprint at a uniform rate.

Updating Task : Each member picks up tasks from the task breakdown and works on them. At the end of the day, they update effort remaining for the task, along with its status.

SAMPLE BURN-DOWN CHART…



PLANNING POKER

Planning poker is a technique which is used to create a Sprint backlog from the Product Backlog.

How It Works :

- To start a poker planning session, the product owner reads a agile user story or describes a feature to the estimators.
- Each estimator is holding a deck of Planning Poker cards with values like 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100.
  - The values represent the number of story points.
- The estimators discuss the feature, asking questions of the product owner as needed.
- When the feature has been fully discussed, each estimator privately selects one card to represent his or her estimate.
- All cards are then revealed at the same time.
  - If all estimators selected the same value, that becomes the estimate.
  - If not, the estimators discuss their estimates and the process will continue until they all agreed for that User story else it will get defer.

НАПОМЕНА

Највећи део материјала ове презентације је преузет из презентације
Scrum Reference Card, аутора Michael James и Luke Walter, која је доступна на адреси:
https://scrumreferencecard.com/ScrumReferenceCard.pdf