*Classical methods of software development have many disadvantages:*

* + *huge effort during the planning phase*
  + *poor requirements conversion in a rapid changing environment*
  + *treatment of staff as a factor of production*

New methods:

Agile Software Development Methodolog

*What is Agile ?*

* *Agile proponents believe*
  + *Current software development processes are too heavyweight or cumbersome*
    - *Too many things are done that are not directly related to software product being produced*
  + *Current software development is too rigid*
    - *Difficulty with incomplete or changing requirements*
    - *Short development cycles (Internet applications)*
  + *More active customer involvement needed*
    - *CMM focuses on process*
* *Agile methods are considered* 
  + *Lightweight*
  + *People-based rather than Plan-based*
* *Several agile methods*
  + *No single agile method*
  + *XP most popular*
* *No single definition*
* *Agile Manifesto closest to a definition*
  + *Set of principles*
  + *Developed by Agile Alliance*

*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*
* *http://www.agilemanifesto.org*

*Agile Methods*

* *Agile methods:*
  + *Scrum*
  + *Extreme Programming*
  + *Adaptive Software Development (ASD)*
  + *Dynamic System Development Method (DSDM)*
  + *…*
* *Agile Alliance (www.****agilealliance****.org)*
  + *A non-profit organization promotes agile development*

*Scrum in 100 words*

*Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.*

*It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).*

*The business sets the priorities. Our teams self-manage to determine the best way to deliver the highest priority features.*

*Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance for another iteration.*

*History of Scrum*

* ***1995:***
  + *analysis of common software development processes* 🡪 *not suitable for empirical, unpredictable and non-repeatable 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:***

*introduction of Scrum at OOPSLA conference*

* ***2001:***

*publication “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*

* *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*
* *Uses generative rules to create an agile environment for delivering projects*
* *One of the “agile processes”*

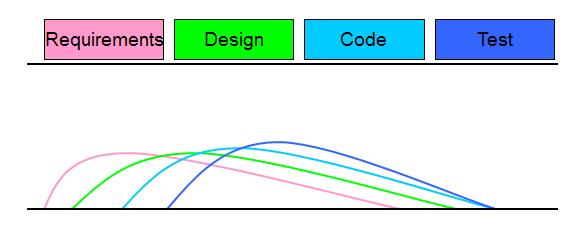
*How Scrum Works?*



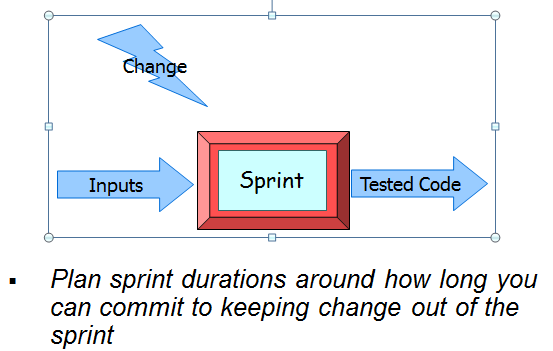
*Sprints*

* *Scrum projects make progress in a series of “sprints”*
  + *Analogous to XP iterations*
* *Target duration is one month*
  + *+/- a week or two*
    - *But, a constant duration leads to a better rhythm*
* *Product is designed, coded, and tested during the sprint*

*Sequential vs. Overlapping Dev.*



*No changes during the sprint*



*Scrum Framework*

* ***Roles*** *: Product Owner, ScrumMaster, Team*
* ***Ceremonies*** *: Sprint Planning, Sprint Review, Sprint Retrospective, & Daily Scrum Meeting*
* ***Artifacts*** *: Product Backlog, Sprint Backlog, and Burndown Chart*

*Product Owner*

* *Define the features of the product*
* *Decide on release date and content*
* *Be responsible for the profitability of the product (ROI)*
* *Prioritize features according to market value*
* *Adjust features and priority every iteration, as needed*
* *Accept or reject work results.*

*The Scrum Master*

* *Represents management to the project*
* *Responsible for enacting Scrum values and practices*
* *Removes impediments*
* *Ensure that the team is fully functional and productive*
* *Enable close cooperation across all roles and functions*
* *Shield the team from external interferences*

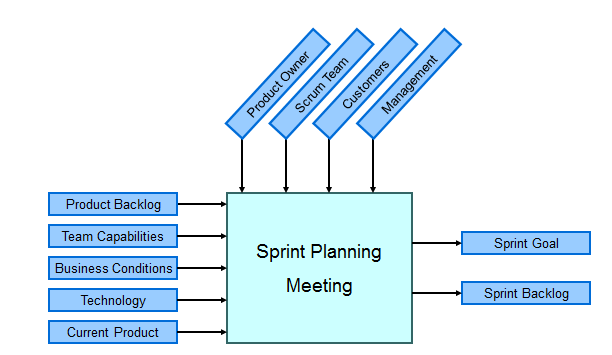
*Scrum Team*

* *Typically 5-10 people*
* *Cross-functional*
  + *QA, Programmers, UI Designers, 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*

*Ceremonies*

* *Sprint Planning Meeting*
* *Sprint*
* *Daily Scrum*
* *Sprint Review Meeting*

*Spring Planning Meeting*



*Parts of 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*

*Pre-Project/Kickoff Meeting*

* *A special form of Sprint Planning Meeting*
* *Meeting before the begin of the Project*

*Sprint*

* *A month-long iteration, during which is incremented a product functionality*
* *NO outside influence can interfere with the Scrum team during the Sprint*
* *Each Sprint begins with the Daily Scrum Meeting*

*Daily Scrum*

* *Parameters*
  + *Daily*
  + *15-minutes*
  + *Stand-up*
  + *Not for problem solving*
* *Three questions:*
  + *What did you do yesterday*
  + *What will you do today?*
  + *What obstacles are in your way?*
* *Chickens and pigs are invited*
  + *Help avoid other unnecessary meetings*
* *Only pigs can talk*
* *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*

*Scrum FAQs*

* *Why daily?*
  + *“How does a project get to be a year late?”*
    - *“One day at a time.”* 
      * *Fred Brooks,* The Mythical Man-Month.
* *Can Scrum meetings be replaced by emailed status reports?*
  + *No*
    - *Entire team sees the whole picture every day*
    - *Create peer pressure to do what you say you’ll do*

*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*

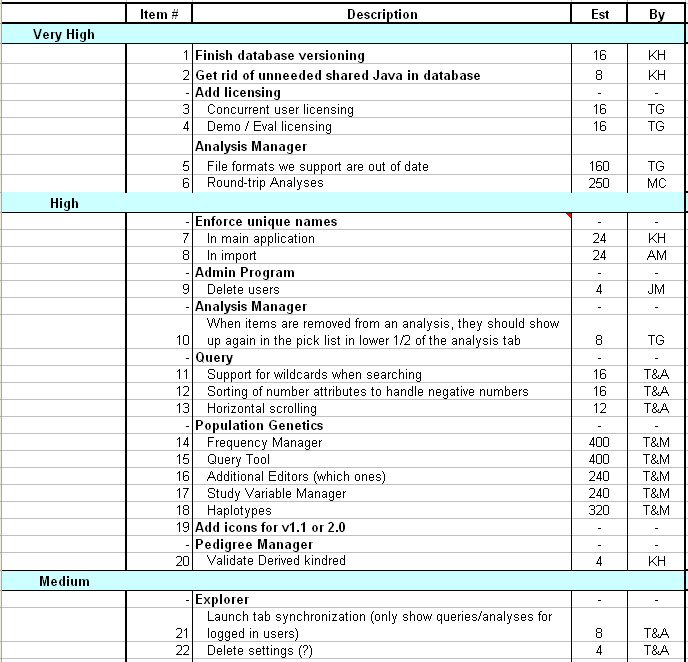
*Sprint Retrospective Meeting*

* *Scrum Team only*
* *Feedback meeting*
* *Three questions*
  + *Start*
  + *Stop*
  + *Continue*
* *Don’t skip for the first 5-6 sprints!!!*

*Product Backlog*

* *A list of all desired work on the project*
  + *Usually a combination of* 
    - *story-based work (“let user search and replace”)*
    - *task-based work (“improve exception handling”)*
* *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*
* *Spreadsheet (typically)*
* *Usually is created during the Sprint Planning Meeting*
* *Can be changed and re-prioritized before each PM*

*Sample Product Backlog*



*From Sprint Goal to Sprint Backlog*

* *Scrum team takes the Sprint Goal and decides what tasks are necessary*
* *Team self-organizes around how they’ll meet the Sprint Goal*
  + *Manager doesn’t assign tasks to individuals*
* *Managers don’t make decisions for the team*
* *Sprint Backlog is created*

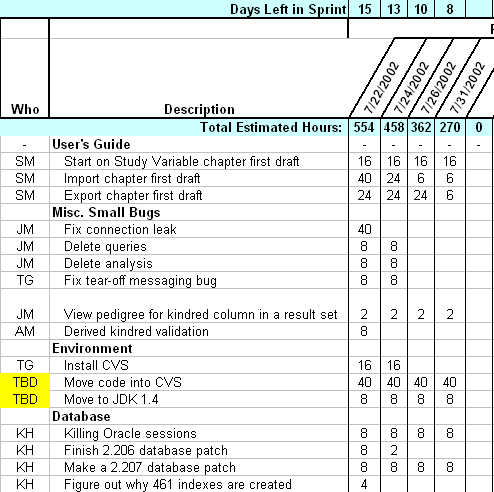
*Sprint Backlog during the Sprint*

* *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*

* *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 do it*

*Sample Sprint Backlog*



*Sprint Burn down Chart*

* *Depicts the total Sprint Backlog hours remaining per day*
* *Shows the estimated amount of time to release*
* *Ideally should burn down to zero to the end of the Sprint*
* *Actually is not a straight line*
* *Can bump UP*

*Information Radiator*

*"Two characteristics are key to a good information radiator. The first is that the information changes over time. This makes it worth a person's while to look at the display... The other characteristic is that it takes very little energy to view the display."*

*Sprint Burndown Chart*



*Release Burndown Chart*

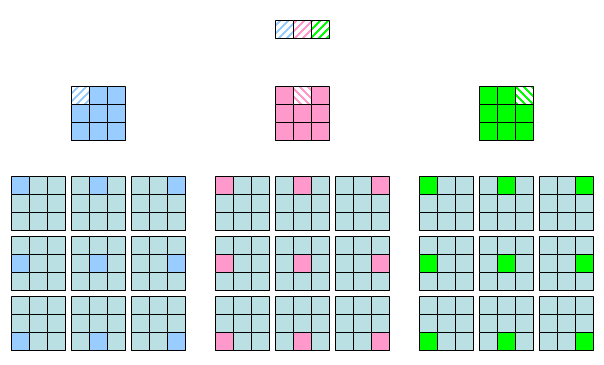
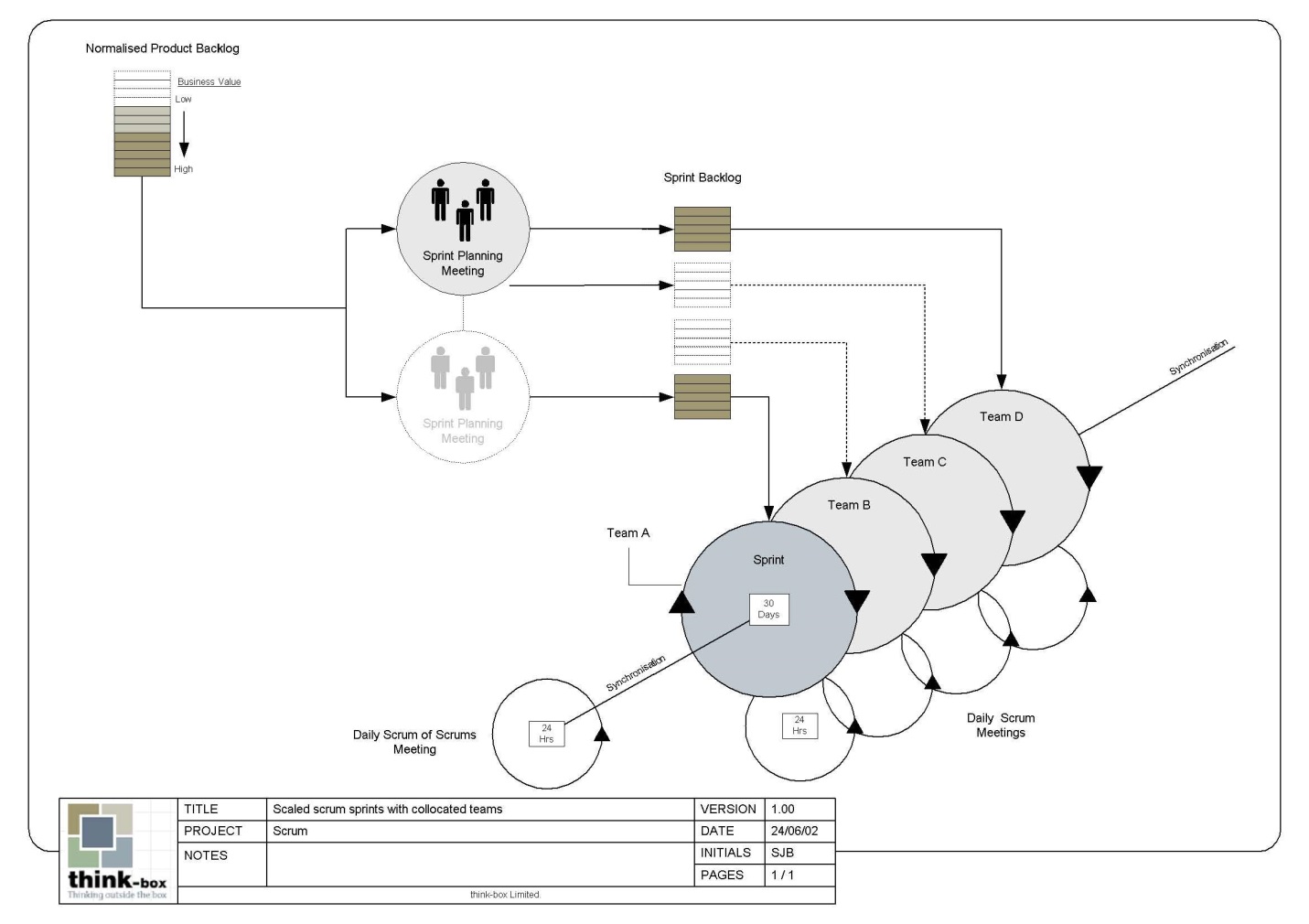
* *Will the release be done on right time?*
* *X-axis: sprints*
* *Y-axis: amount of hours remaining*
* *The estimated work remaining can also burn up*

*Product Burndown Chart*

* *Is a “big picture” view of project’s progress (all the releases)*

*Scalability of Scrum*

* *A typical Scrum team is 6-10 people*
* *Jeff Sutherland - up to over 800 people*
* *"Scrum of Scrums" or what called "Meta-Scrum“*
* *Frequency of meetings is based on the degree of coupling between packets*



*Pros/Cons*

