Scrum Methodology

Presented by Simon Baker

think-box Limited

http://www.think-box.co.uk

Agenda

- Agile software development
- A quick tour of Scrum
- Scrum in more detail
- QA

Agile software development

delivering the right software at the right time

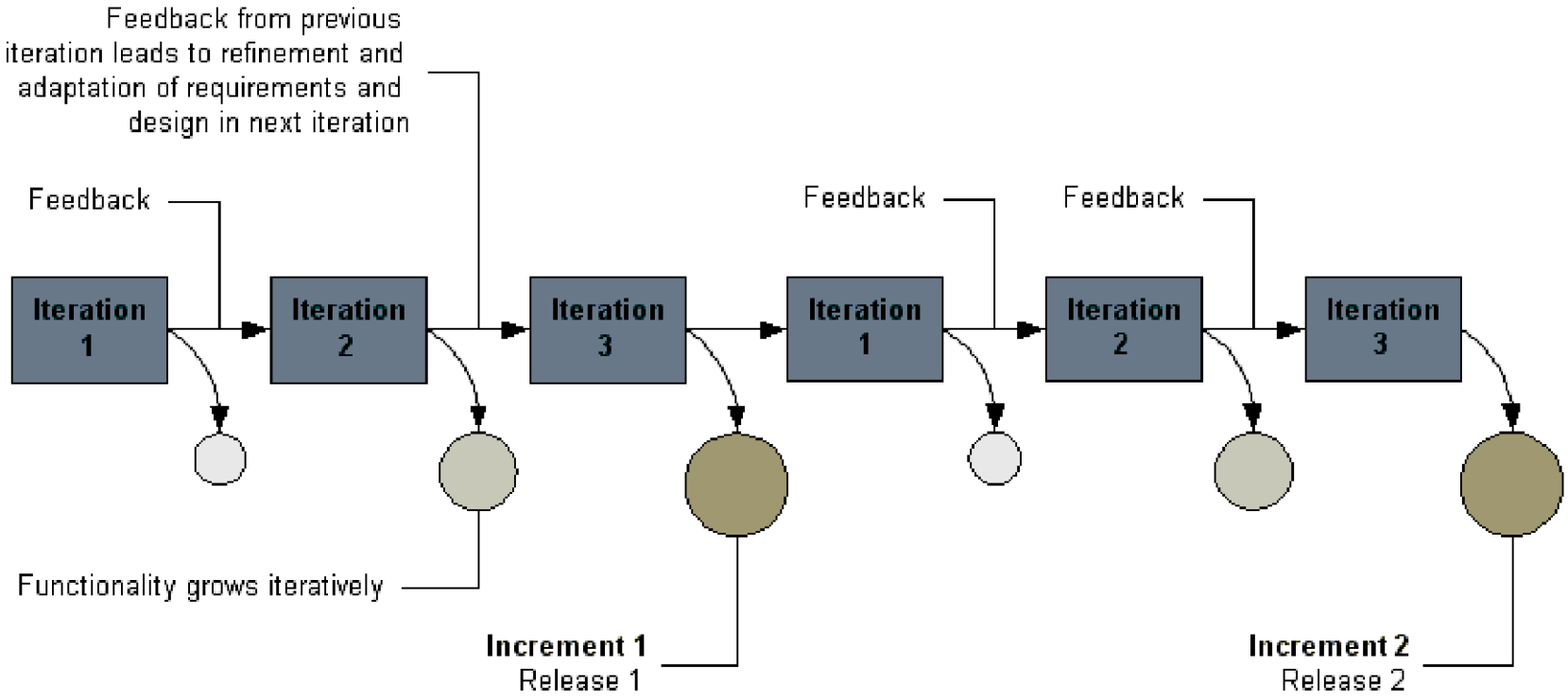
Agile characteristics

- Manoeuvrable through the application of time-boxed iterative development and adaptive planning while promoting evolutionary delivery
- Steering through early and frequent feedback
- Feedback
  - » Continuous testing
  - » Continuous integration
  - » Continuous review

Complementary strategies

- Evolutionary
  - » The requirements, plan and estimates all evolve over time
  - » They are not fully defined and frozen in a major up-front effort
- Incremental
  - » Software is delivered in a series of releases with expanding capabilities
- Iterative
  - » Make progress through successive refinement
  - » With each iteration software is improved through the addition of greater detail
  - » Delivers completed, production-quality code every iteration

Iterative and incremental



Time-boxed

- The iteration duration is fixed
- Never slip the iteration end date
  - » People remember slipped dates, not slipped features
- If the iteration contents cannot be delivered within the iteration then de-scope some lower business value user stories to next iteration
- Once the iteration has started the content cannot be changed, unless:
  - » Content has been completed early and more work is required
  - » Content has not been completed and user stories need to be de-scoped
- Benefits:
  - » Maintains focus
  - » Forced to tackle small levels of complexity
  - » Forced to make difficult decisions and trade-offs early

Adaptive planning

The Customer adaptively plans the content of the next iteration shortly before it starts, based on:

- Business situation or changing priorities
- Latest insight
- Feedback from users, tests, developers, etc

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

While there is value in the items on the right we value the items on the left more

See http://www.agilealliance.com, http://www.agilemanifesto.org

Agile principles [1/2]

- Highest priority is to satisfy the Customer through early and continuous delivery of valuable software
- Welcome changing requirements and harness change for the Customer's competitive advantage
- Deliver completed software frequently and regularly
- Business people and developers work collaboratively throughout the project
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done
- Most efficient and effective method of conveying information to and within a team is face-to-face conversation

Agile principles [2/2]

- Working software is the primary measure of progress
- Developers should be able to sustain pace indefinitely
- Continuous attention to technical excellence and good design enhances agility
- Simplicity – the art of maximising the amount of work not done – is essential
- At regular intervals, teams should reflect on how to become more effective, then tune and adjust behaviour accordingly

Agile benefits [1/2]

- Early risk mitigation and discovery
  - » Risks emerge progressively and perceived risks prove not to be
  - » Unsuspected issues are forced into the open
- Iterations accommodate and provoke early change
- Manageable complexity
  - » Iterative development breaks complex projects into small bounded mini-projects
- Early partial product
  - » Early visible progress with integrated and tested code
  - » Gives developers confidence and Customer confidence in development team
- Relevant progress tracking and better predictability
  - » Each iteration delivers completed software
  - » Tracking ‘actual effort’ sets workload-limit for next iteration (Velocity)

Agile benefits [2/2]

- Higher quality code with less defects
  - » Test-driven development
- Better chance of delivered software matching Customer's desires
  - » Early and continuous evaluation and assessment of software
- Early and regular process improvement
  - » Retrospective assessment after each iteration
- Communication and engagement
  - » Iterations force early integration, co-ordination and communication
  - » Iteration planning engages Customer
  - » Feedback from Customer increases their engagement
  - » Per-iteration demonstration requires the presence of Customer for sign-off

Agenda

- ☑ Agile software development
- ☐ A quick tour of Scrum
- ☐ Scrum in detail
- ☐ QA

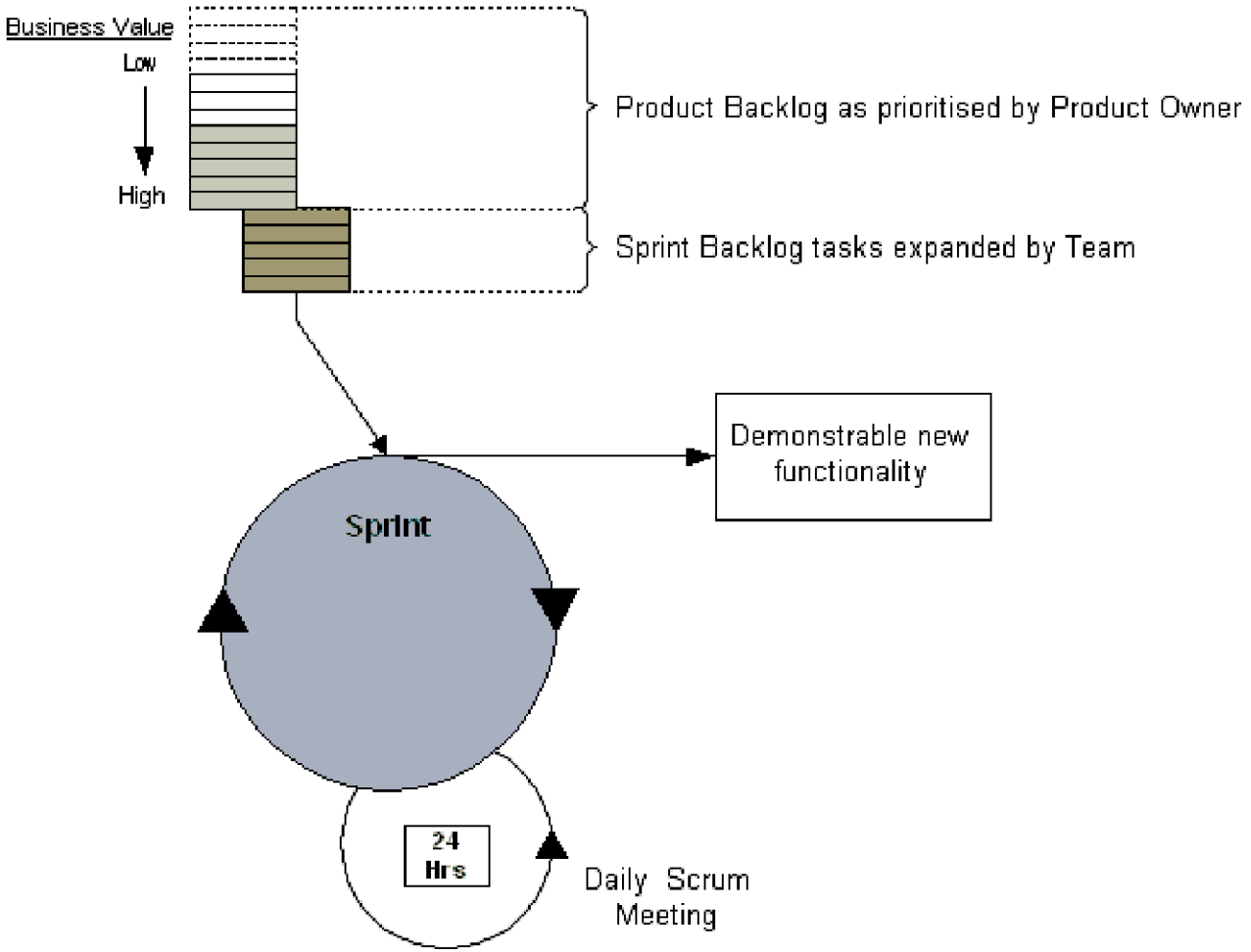
A quick tour of Scrum

a collaborative methodology designed
to capitalise on peoples' strengths

Overview [1/2]

- Simple and scaleable (has been used in teams of 600+)
- Based on an empirical process
- Ideally suited for projects with rapidly changing or emergent requirements
- Empowers and focuses developers
- Provides an environment in which software can be built quickly
- Does not prescribe how to elicit/represent requirements
- Does not prescribe how to develop software

Overview [2/2]



Empirical process

- Visibility
  - » Make everything visible
  - » What is visible must also be true
  - » The process must be visible to those controlling the process
- Inspection
  - » The process and resulting software must be inspected frequently
- Adaptation
  - » Adjustment must be made quickly to minimise further deviation

Scrum characteristics

- Simple practices and management work products
- Individual and team problem solving and self-management
- Evolutionary, iterative and incremental requirements and development
- Adaptive behaviour
- Customer participation and steering
- Openness and visibility
- Easily combined with other methods
- Communication and learning

Agenda

- ☑ Agile software development
- ☑ A quick tour of Scrum
- ☐ Scrum in more detail
- ☐ QA

Scrum in more detail

inspect and adapt

Product Backlog

Each product line is represented by a Product Backlog

- An effective mechanism for directing product lines, Sprint by Sprint, to deliver increments of functionality with the highest business value
- An evolving queue of work for a product line
- Owned by a Product Owner
- Prioritized by business value
  - » Top priority Product Backlog items drive analysis and development activities
  - » Reprioritization occurs in response to changes in the business

Product Owner

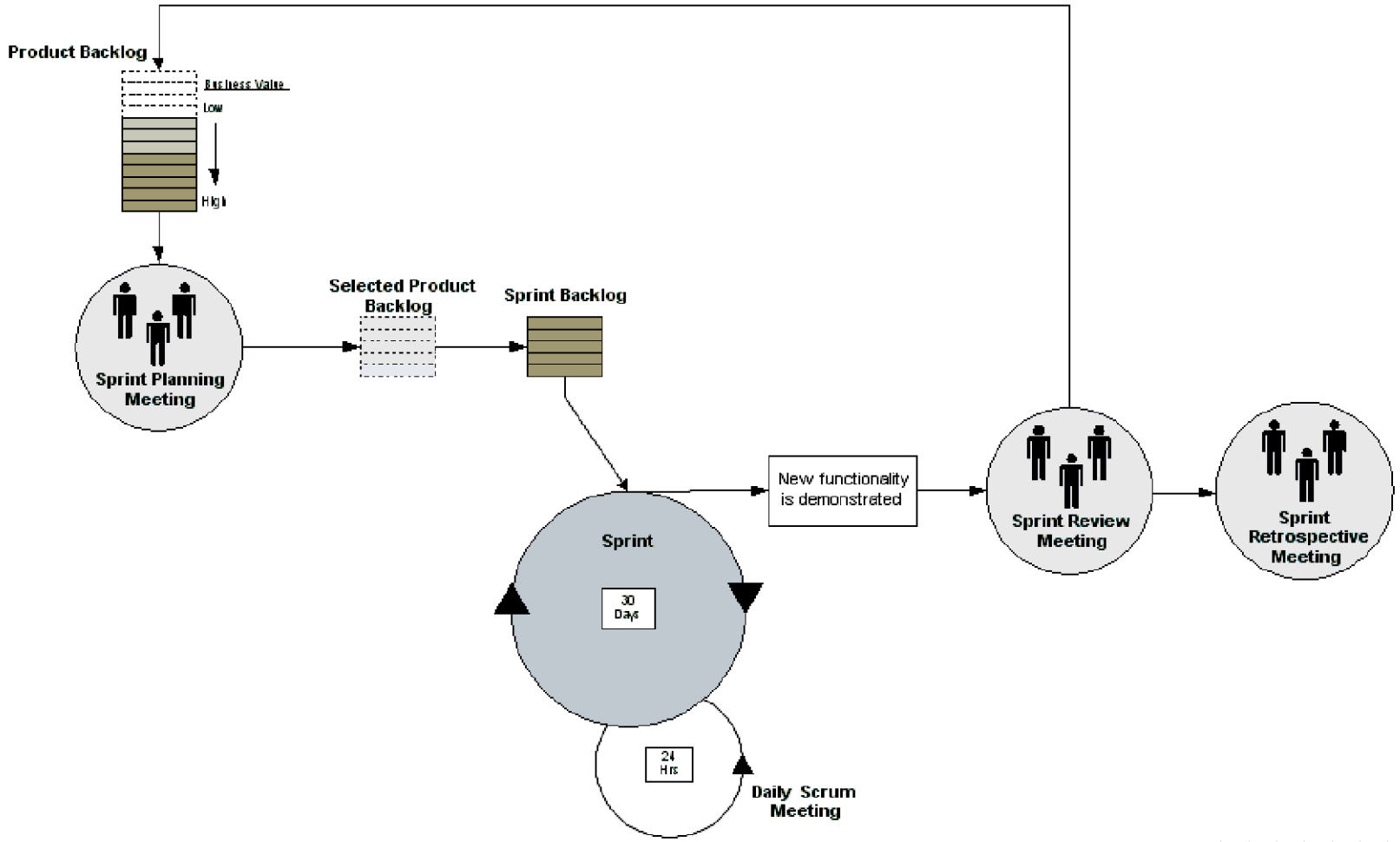
- A single business contact for each product with the authority to represent the Business to the Scrum Team
- Owns the Product Backlog
  - » Authority to quantify and qualify business requests
  - » Authority to manage and prioritize the Product Backlog to ensure functionality with the highest business value is delivered first
- Selects the functionality to be developed during each Sprint
- Works with the Scrum Team to ensure requirements are well understood
- Continuously engaged throughout development to provide feedback
- Reviews the delivered software (with other business stakeholders)

Scrum Master

A facilitator serving a Scrum Team

- Keeps the Scrum Team moving at a sustained pace by:
  - » Makes decisions ideally immediately or within an hour
  - » Removes impediments before the next Daily Scrum
  - » Relays up-to-date progress information to management, keeping things visible
  - » Enforces Scrum values and practices
- Improves the lives of the Scrum Team by facilitating creativity and empowerment
- Helps the Product Owners drive development directly
- Mediates between Management and the Scrum Team
- Helps plan the Sprint and reinforces the Sprint goal

Lifecycle



Sprint Backlog

- Represents the work that the Scrum Team commits to completing in the Sprint
- Contains Product Backlog items selected for the Sprint
- Scrum Team updates the public Sprint Backlog daily to reflect progress:
  - » Expended effort
  - » Remaining effort
  - » Completed tasks
- How does the Sprint Backlog change during the Sprint?
  - » Scrum Team adds new tasks whenever needed in order to meet Sprint Goal
  - » Scrum Team can remove unnecessary tasks
  - » Estimates are updated whenever there's new information
  - » Sprint Backlog can only be updated by the Scrum Team

Scrum Team

- Their purpose is to convert Product Backlog into a working increment of functionality that satisfies the Sprint goal
- Self-organising
- Members should be full-time; membership can change only between Sprints
- Make development decisions and manage development activities
- Define the engineering work in the Sprint
- Conform to existing standards, conventions, technologies, etc

Sprint Planning Meeting [1/2]

- Held at the start of each Sprint
- Typically lasts about a day
- The Product Owner, Scrum Master and Scrum Team attend
- The purpose if to define the Sprint Backlog and plan the Sprint
- The Scrum Team commits to achieving the Sprint goal
- The Product Owner commits that Sprint contents will not be changed for duration of the Sprint

Sprint Planning Meeting [2/2]

- Demonstrates adaptive planning driven by the Product Owner
- The Sprint has started and clock is ticking
- First half:
  - » Product Owner describes highest priority items on Normalised Product Backlog
  - » Scrum Team provides high-level estimates
  - » Product Owner/Scrum Team selects items for Sprint
  - » Scrum Team moves selected items to Sprint Backlog
  - » Scrum Team define Sprint Goal – a theme for the Sprint
- Second half:
  - » Scrum Team disaggregate user stories into engineering tasks and estimate

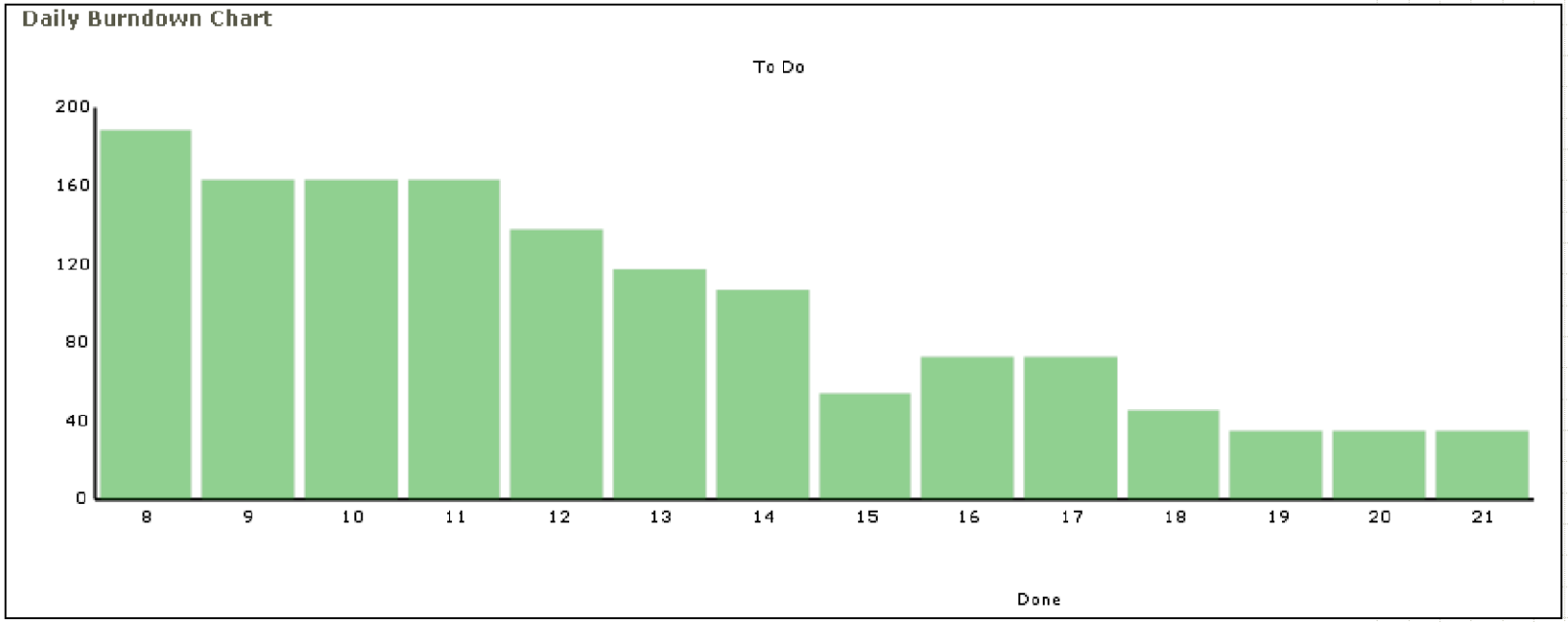
Sprint [1/2]

- Scrum projects make progress in a series of Sprints
- Duration can be governed by:
  - » Amount of time required to build something valuable for Product Owner
  - » Maximum time most stakeholders will wait without losing interest
  - » Maximum time most stakeholders will wait before they can request more work
- The Sprint duration must be constant to establish a rhythm
- Work cannot be added once the Sprint is underway
- Uninterrupted focus is maintained
- The Sprint must deliver completed software

Sprint [2/2]

Sprint Burndown Chart:

- Visual summary of engineering tasks in hours remaining in the Sprint Backlog
- Critical tracking data



Daily Scrum [1/2]

- Held at the same time each morning
- Typically 15 minutes in duration
- The Scrum Master, Scrum Team and Product Owner attend
- Each team member answers the questions:
  - » What did you achieve since yesterday?
  - » What are you aiming to achieve today?
  - » What obstacles are in your way and what dependencies do you have?
- When a person reports on what they are doing for the next day, they are expressing a kind of social promise to the team

Daily Scrum [2/2]

- It is the heartbeat of Scrum and provides a daily status snapshot
  - » Frequent measuring and adaptive response mechanism
  - » Opportunity for Scrum Team to assess and reconsider assignments
  - » Forum from which to update tasks and remove impediments
- Allows resolution of dependencies and conflicts in real time to maximise productivity
- 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 Scrum Team sees the whole picture every day

Sprint Review

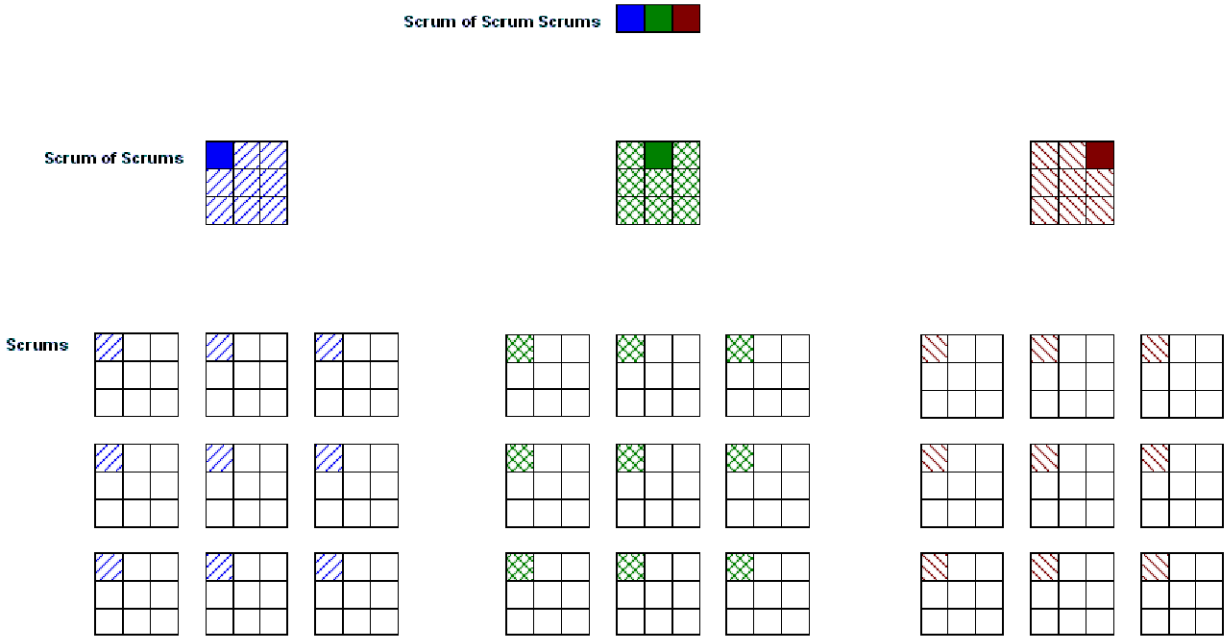
- Held at the end of each Sprint to establish whether the deliverable has satisfied the Sprint Goal
- A maximum of 4 hours duration
- Product Owner, Scrum Master and the Scrum Team attend
- Demonstrate business functionality and acceptance tests passing
- Opportunity for Product Owner to inspect functionality and make timely adaptations to project
- Rules:
  - » Informal – no PowerPoint presentations
  - » 2-hour prep time rule
  - » Should not be a distraction for Scrum Team
  - » Should be a natural result of Sprint

Sprint Retrospective

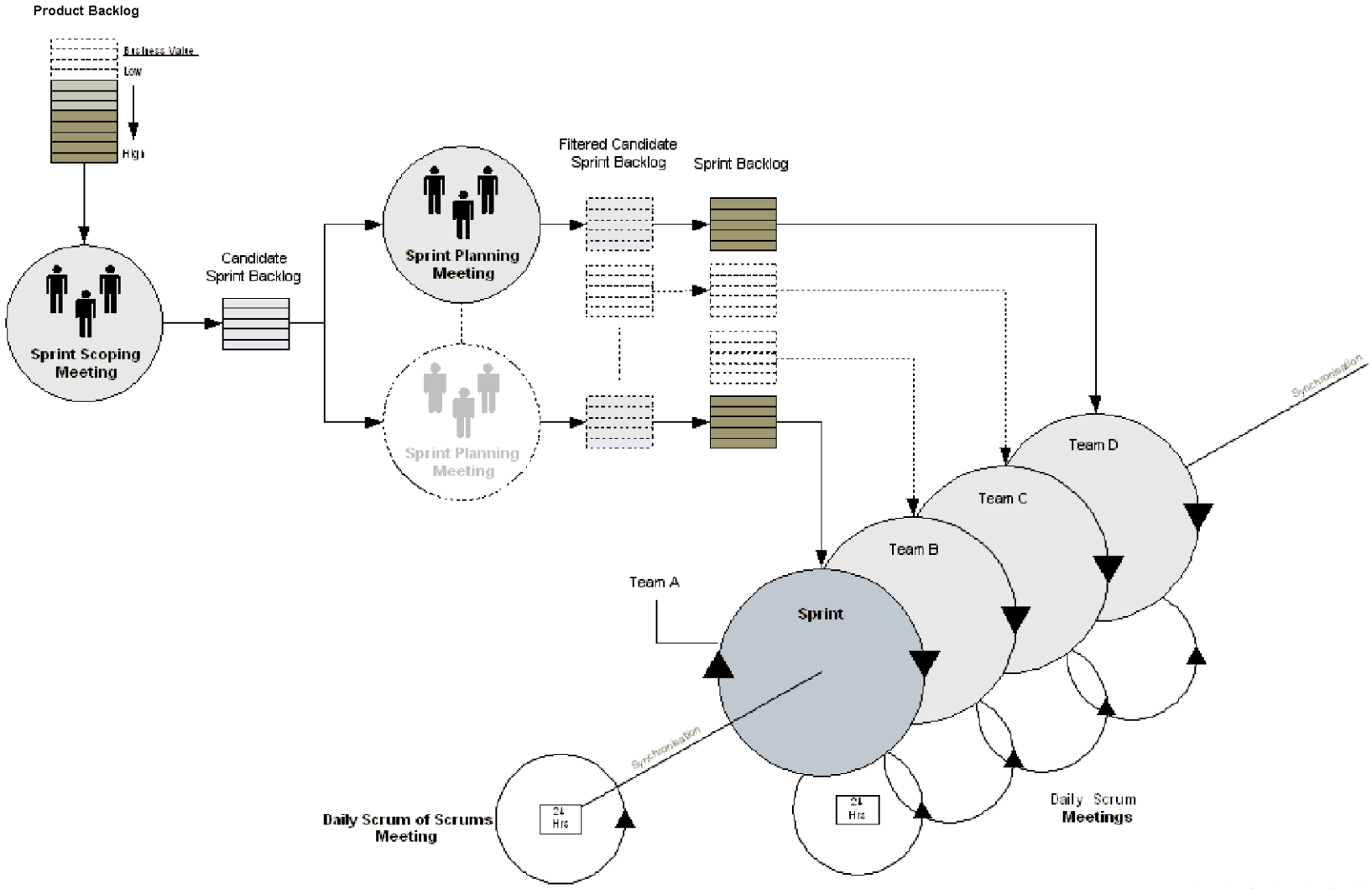
- Held after the Sprint Review
- A maximum of 2 hours duration
- Product Owner, Scrum Master and the Scrum Team attend
- Purpose is to analyse how the Sprint went
- Opportunity to revise the development process within the Scrum framework and practices

Scaling Scrum [1/2]

- The primary way of scaling Scrum to work with multiple teams is to coordinate a Daily Scrum of Scrums
- Each team contributes the Scrum Master and Team Lead to a regular Scrum of Scrums to coordinate and manage inter-team dependencies within the Sprint



Scaling Scrum [2/2]



Scrum of Scrums

- Product Owner and Scrum Masters or representatives from the Scrum Teams attend
- Purpose is to communicate progress and impediments among all teams and to measure impact and adjust the Sprint accordingly
- Each person answers the questions:
  - » What did your team achieve since yesterday?
  - » What is your team aiming to achieve today?
  - » What obstacles are in your team's way and what dependencies do they have?

Agenda

- ☑ Agile software development
- ☑ A quick tour of Scrum
- ☑ Scrum in more detail
- ☐ QA

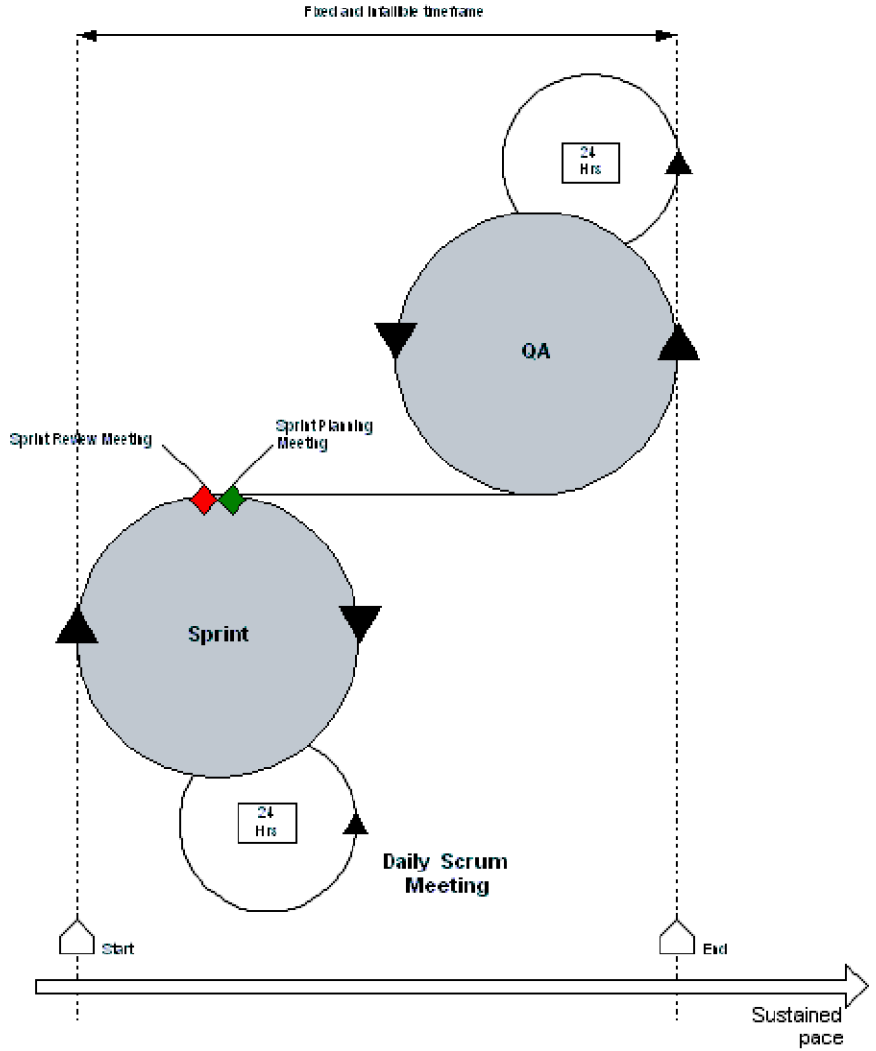
QA

delivering quality software

Stabilization Sprints

- Use stabilization sprints:
  - » To prepare for release
  - » During active beta periods
  - » When transitioning a team to Scrum
  - » When quality isn't quite where it should be

QA uses a stabilization Sprint

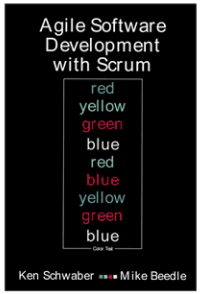


Agenda

- ☑ Agile software development
- ☑ A quick tour of Scrum
- ☑ Scrum in more detail
- ☑ QA

References

- *Agile Software Development with Scrum*
  Schwaber K (2002), Prentice Hall



- *Agile Project Management with Scrum*
  Schwaber K (2003), Microsoft