How we're Doing Agile Development at CA

Simon Cockayne
Principal Software Engineer & ScrumMaster
CA 2E Development Team

GENERAL SESSION IN SALON E-F
Friday June 3rd 2011 (1:15pm -2:00pm)

Legal

This presentation was based on current information and resource allocations as of **February 02, 2011** and is subject to change or withdrawal by CA at any time without notice. Notwithstanding anything in this presentation to the contrary, this presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion. Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA will make such release available (i) for sale to new licensees of such product; and (ii) to existing licensees of such product on a when and if-available basis as part of CA maintenance and support, and in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis. In the event of a conflict between the terms of this paragraph and any other information contained in this presentation, the terms of this paragraph shall govern.

Legal

CERTAIN INFORMATION IN THIS PRESENTATION MAY OUTLINE CA'S GENERAL PRODUCT DIRECTION. ALL INFORMATION IN THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY AND MAY NOT BE INCORPORATED INTO ANY CONTRACT. CA ASSUMES NO RESPONSIBILITY FOR THE ACCURACY OR COMPLETENESS OF THE INFORMATION. TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT WILL CA BE LIABLE FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENT, INCLUDING, WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Abstract

CA 2E r8.6 is being developed using an Agile, scrum-based, approach. This session will explain how Agile is helping the 2E team give customers the new features they need, in a shorter timeframe, packed into a higher quality, more robust release. You'll see not only how 2E customers and partners, can literally become part of the 2E development and evaluation process…but also learn valuable lessons to apply to your own Agile projects. You'll also see that this session abstract is not just a bag of hip buzzwords…but real facts based on real experience.

Agenda

- What is Agile?
- What is Scrum?
- How to build a brilliant Product Backlog using CA Agile Vision
- User stories, story points and planning poker
- Pre-release testing
- Useful resources
- Q&A

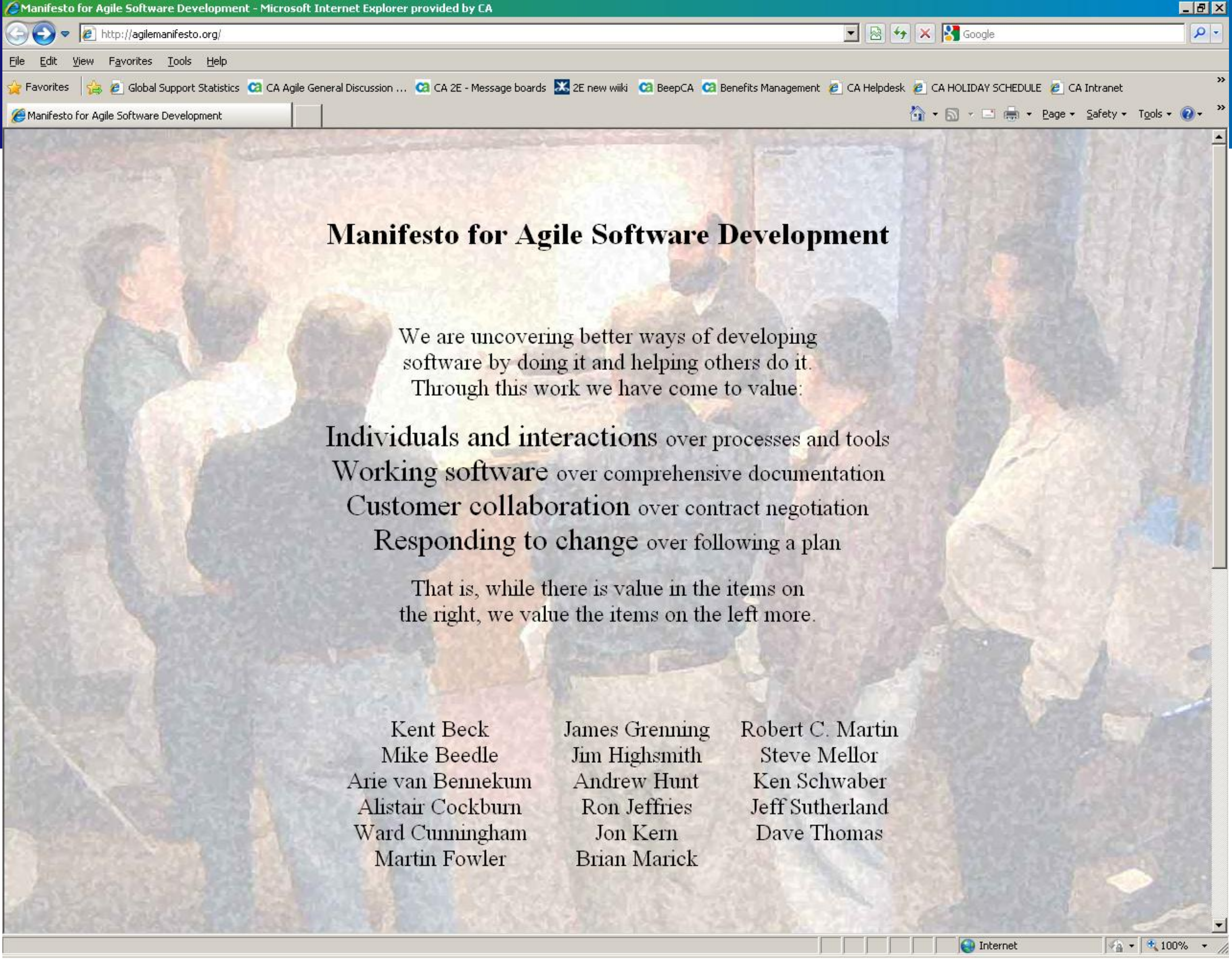
What is Agile?

Agile overview

WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
scottadams@aol.com
www.dilbert.com
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
11-26-07 ©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.

via Mario Moreira


Manifesto for Agile Software Development - Microsoft Internet Explorer provided by CA
http://agilemanifesto.org/
Google
File
Edit
View
Favorites
Tools
Help
Favorites
Global Support Statistics
CA Agile General Discussion ...
CA 2E - Message boards
2E new wiki
BeepCA
Benefits Management
CA Helpdesk
CA HOLIDAY SCHEDULE
CA Intranet
Manifesto for Agile Software Development
Page
Safety
Tools
Internet
100%

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 overview

A set of Principles and Values, based on the Agile Manifesto, that focus on the delivery of value to the customer

An umbrella term for a collection of Agile methodologies, practices, and techniques

Based on iterative and incremental approaches where requirements and solutions evolve through collaboration between self-organized teams and customers

Aligned with Lean Principles of eliminating waste and increasing efficiency and productivity

Not just for engineers but for the whole Product Team

via Mario Moreira

Principles behind the Agile Manifesto

"We follow these principles:

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.

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.

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." (http://agilemanifesto.org/)

What is scrum?

What is Scrum?

- *“Scrum is an iterative, incremental process for developing any product or managing any work. It produces a potentially shippable set of functionality at the end of every iteration. It's attributes are:*
  - *Team-based approach to develop systems and products when requirements are rapidly changing*
  - *Controls the chaos of conflicting interests & needs*
  - *Improves communications and maximize co-operation*
  - *Maximize productivity*
  - *A way for everyone to feel good about their job, their contributions, and that they have done the very best they possibly could”*

Controlchaos.com

Where did the term Scrum come from?

“The ‘relay race’ approach to product development…may conflict with the goals of maximum speed and flexibility. Instead a holistic or ‘rugby’ approach – where a team tries to go the distance as a unit, passing the ball back and forth – may better serve today’s competitive requirements.”

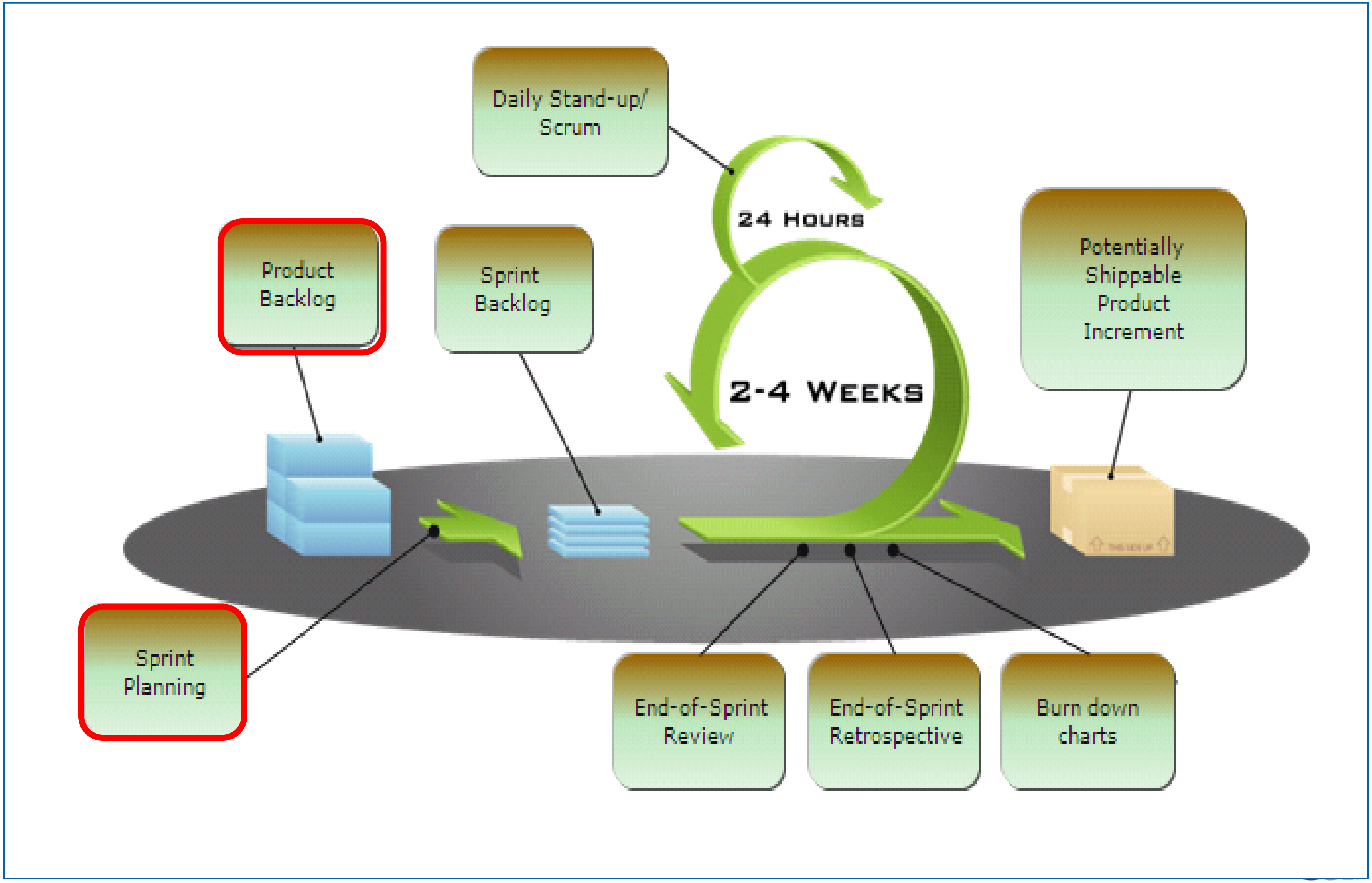
Hirotaka Takeuchi and Ikujiro Nonaka,
“The New New Product Development Game”,
Harvard Business Review, January 1986.

Image via Mario Moreira

Roles in Agile Scrum Team

- **ScrumMaster**
  - Responsible for teaching & implementing Scrum method and practices
  - Acts as a **facilitator** (not the team leader, not the project manager)
  - Facilitates Daily Scrum and other meetings
  - Help remove roadblocks
- **Scrum Team**
  - Typically about 7-12 people
  - Cross-functional (programmers, test, design, CM, tech writer)
  - Members should be full-time (committed)
  - Team should be self-organizing
- **Product Owner**
  - Voice of the Customer
  - Manage Product Backlog
  - Prioritizes the requirements/stories (Sprint Planning)
  - Validates the functionality delivered in sprint (End-of-Sprint Reviews)

Sprint-based development



Daily Scrum

- **Is a <u>very</u> short time-boxed meeting** (15 minutes)
- **A <u>daily</u> meeting** for ScrumMaster and Scrum Team (Pigs)
  - **Chickens can listen…but should not speak.**
- **<u>Every team member should answer 3 questions</u>**
  - **1) What did you do since the last Scrum and is anything 'done'?**
    - 'Done' means 'meets the definition of done' for a backlog item (a.k.a. user story).
  - **2) What will you do until the next Scrum?**
  - **3) Do you have any roadblocks preventing you from doing your work?**

Daily Scrum tips

BE ON TIME.

SET AN ALARM FOR FIFTEEN MINUTES.

DON’T GO INTO DETAILS FOR OFF-TOPIC ITEMS.

DO NOT BE SHY IN TELLING PEOPLE TO STICK TO THE POINT.

DO NOT SOLVE PROBLEMS DURING THE STANDUP.

ANNOUNCE THE END OF THE STAND-UP

(TELL PEOPLE THEY CAN LEAVE IF THEY WANT)

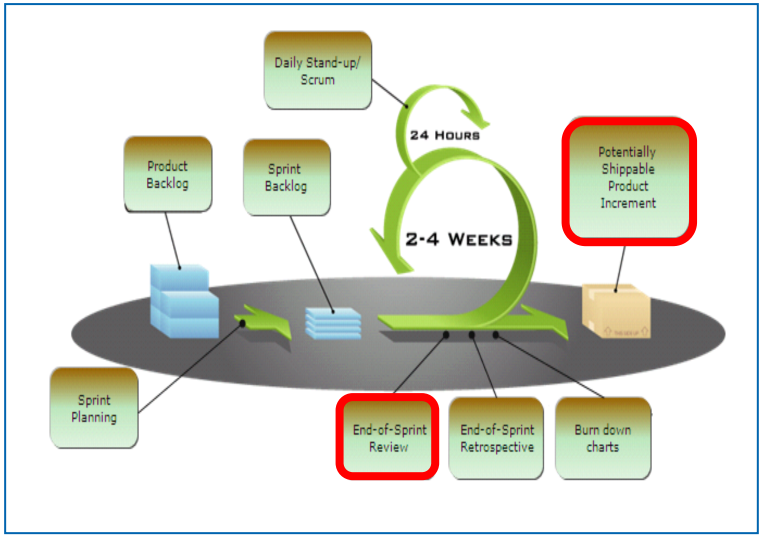
DISCUSSIONS CAN CONTINUE AT THE END OF THE CALL

(BUT THE STANDUP IS DONE).

DO NOT point fingers or play the blame-game.

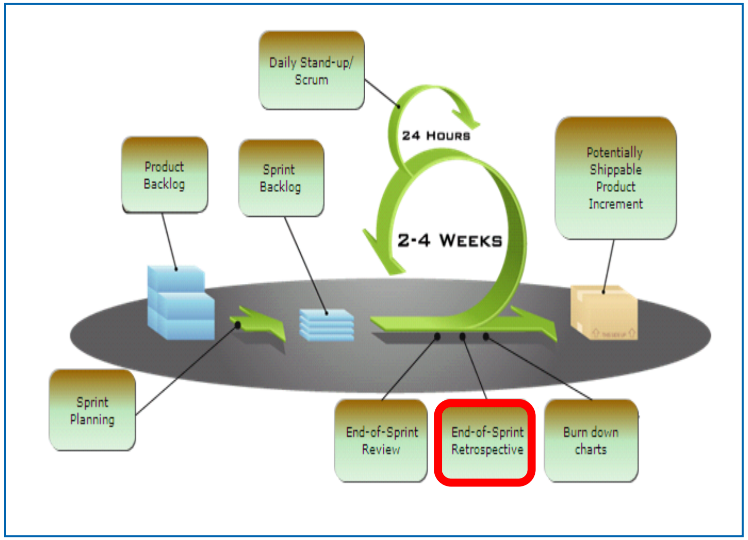
End-of-Sprint Review

- Typically takes the form of a demo of new features or underlying architecture
- Team presents what it accomplished during the sprint
  - No software can be demonstrated to the Customer and Product Owner that is not “done”.
- Participants
  - Customers
  - Management
  - Product Owner
  - Other engineers
- What are you gaining from this session?
  - Look for feedback of requirements
  - Does it meet Acceptance Criteria
  - Is it ready for release
  - Reduce uncertainty

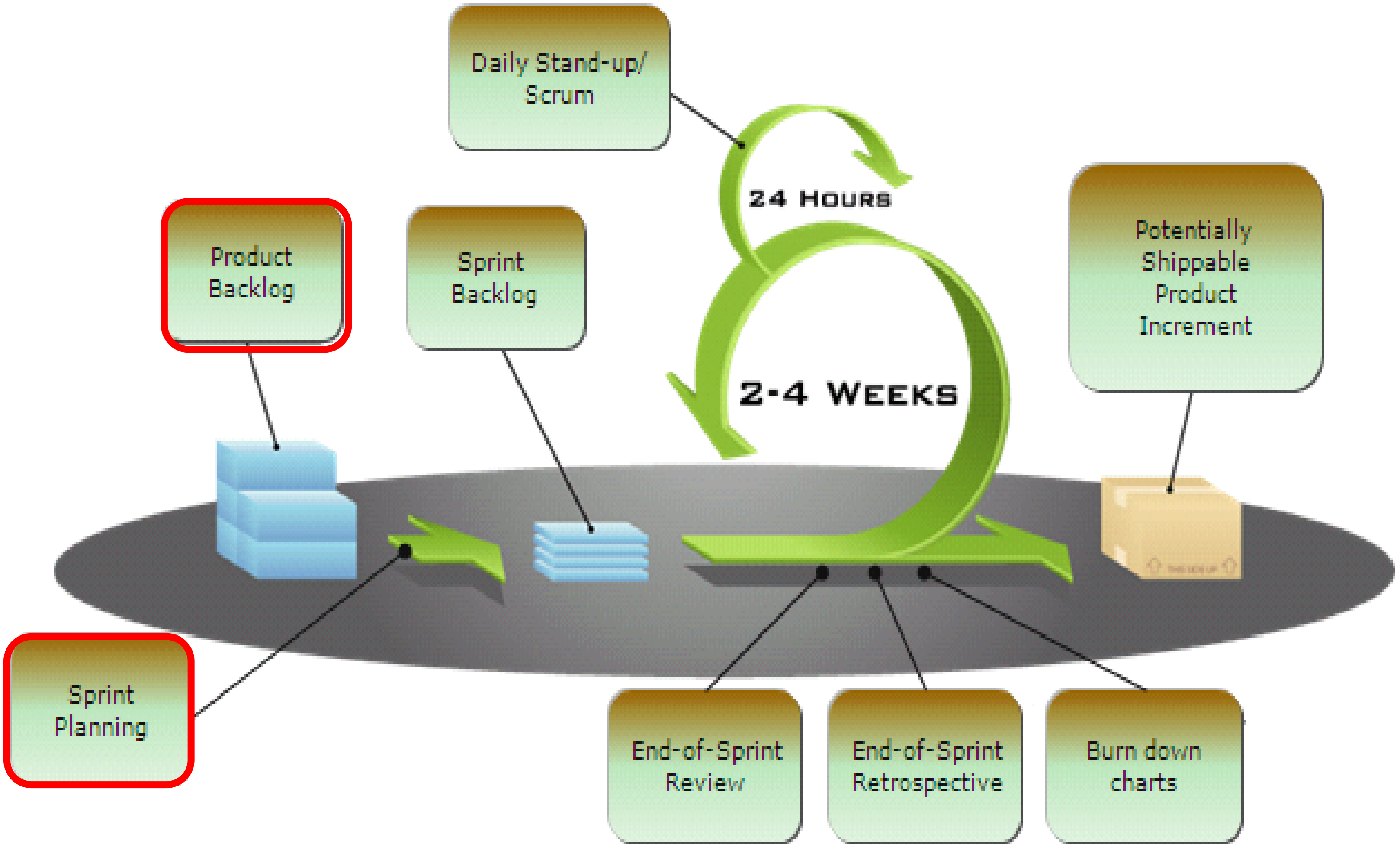


Sprint Retrospective

- Determine:
  - What went well
  - What can be improved
- From the “improvement” list, select only 1 or 2 things that are prioritized as the high value.
- Estimate effort of doing improvement task(s) and add to next sprint backlog.
- Participants
  - With just the Scrum Team
  - PO may participate



Sprint-based development



Agenda

How to build a brilliant Product Backlog using CA Agile Vision

What is the Product Backlog?

- Product Backlog is a “wishlist” of requirements for the product that the scrum team is working on.

- Each requirement is known as a Backlog Item, or User Story

- Product Owner is responsible for maintaining (grooming) the backlog

- ‘Grooming’ includes: adding, removing, changing and prioritizing user stories.

- Includes enhancements and bug fixes

- Product backlog is prioritized by ‘business value’.

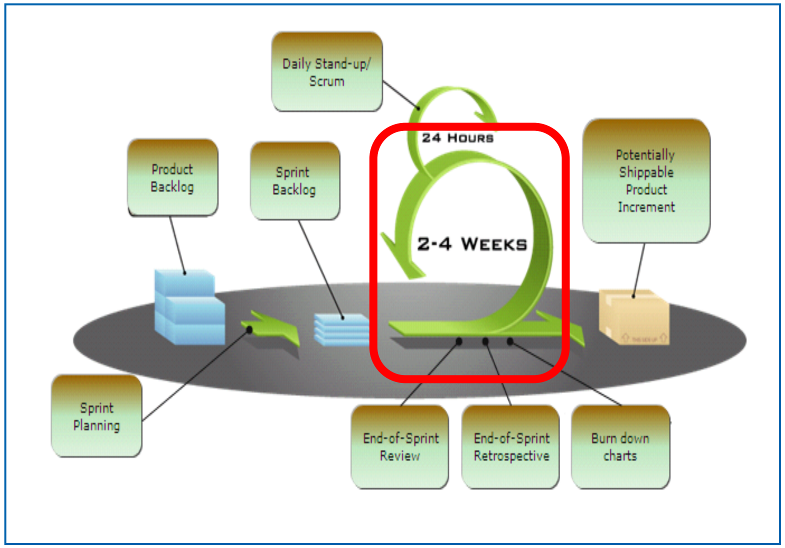
- **The Product Backlog is dynamic.** It is never finished.

What is the Sprint Backlog?

- Sprint Backlog is a subset of the Product Backlog.
- During a sprint a scrum team will self-organize to complete all the stories in the Sprint Backlog.

Sprint

- An short time-boxed iteration of work (e.g. 4 weeks) during which scrum team **self-organizes** to complete all stories in Sprint Backlog
- Goal is producing an increment of potentially shippable code at the end of each sprint
- No outside influence should interfere with the team's progress during a sprint
- No changes should be made to the requirements/stories during a sprint
  - Changes can be introduced at the next Spring Planning session



So...*come on then*...What is a Story?

- A story is a requirement.

- Use canonical form

As a <user role/persona>, **I want to** <action/behavior>, **so that** <business value>

E.g:
As a user of a 2E model **I want to have a** function key added to the 2E logo 'splash screen' to display a source member or equivalent containing details of 'What's new?' in the product at the new release, **so that** I can be introduced to and become excited about new features.

But where is the detail???

* Detail is in the **conversation**.

  Captured in….

  - Tasks
  - Acceptance Criteria
  - Comments (Optional)
  - Design docs (Optional)

Tasks

* Specific pieces of work that need to be performed.
* Scrum team members will commit to tasks during Sprint Planning or during Sprint

2E team uses generic tasks…

- Design
- Code and unit test
- Write Test Cases
- Execute Test Cases
- DOC
- DOC Verification (all scrum team members)

… and a template story:

Acceptance Criteria

The specific criteria that need to be met before a story can be closed.
QA can use acceptance criteria to design and build test cases.

QA can suggest acceptance criteria…so can ***anyone*** in the scrum team.

e.g:

The 2E splash screen should allow the user the option to go straight into the model, or to display a 'What's New' source member, before entering the model.

Don't repeat the "Definition of done" in acceptance criteria.
Don't duplicate all your Tasks in Acceptance Criteria

Tips for using Agile Vision

Right-click to work in new tab.

Create ‘template’ stories and clone to make new ones.

- Tasks can be copied.
- Acceptance criteria need to be recreated.

Update task worklogs, and tasks and acceptance criteria EVERY DAY.

In group meetings, make notes in word, then do all updates after the meeting.

Reorder by changing order number fields (don’t use drag and drop)

Comments and design docs

Can be added to stories

How to add a story to the Backlog

- Define the, canonical, definition.
- Create a new story (or copy from a template story).
- Add known Acceptance Criteria.
- If you don’t know the priority, make it “Not Prioritized”.
- Add/Modify tasks (do NOT assign or estimate).
- Do NOT assign the story to anyone.
- Is story targeted for a given release?
- Has the team estimated the size of the story?
    - Can be done in Backlog grooming.
- Move the story into the correct position in the backlog.

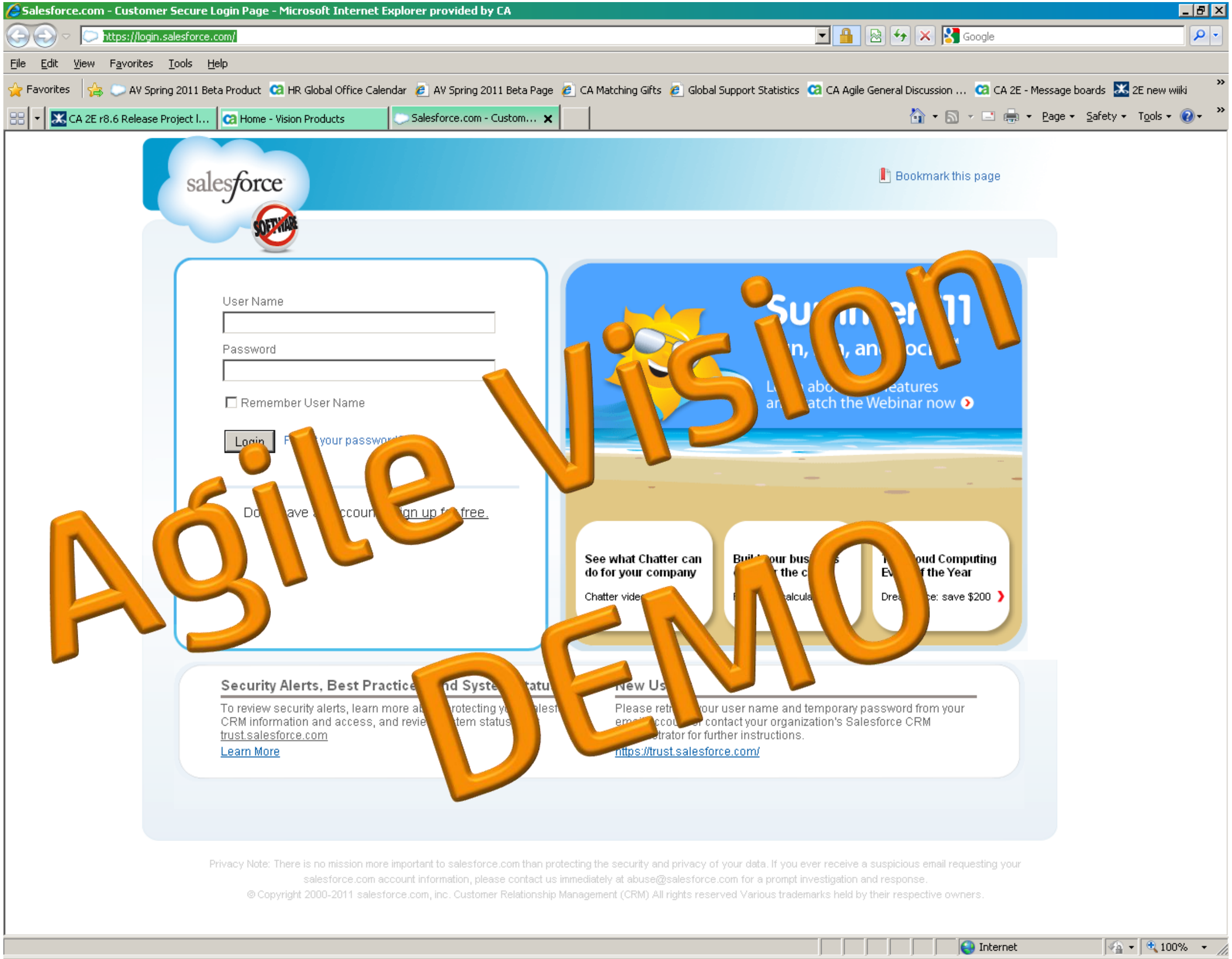
- TIPS: Have one person responsible for adding stories and grooming backlog.
    - Ideally the PO (Product Owner)…alternatively ScrumMaster

Correct position in Backlog

- Template stories at the top.
- [TASK] Stories
- Prioritized stories:
    - Priority
    - Release
    - Sprint

    NOTE: Can also group per-sprint stories
- “Not Prioritized” stories
- [MERGED] stories
- [Closed] stories at the bottom.

https://login.salesforce.com/


Salesforce.com - Customer Secure Login Page - Microsoft Internet Explorer provided by CA
https://login.salesforce.com/
Google
File
Edit
View
Favorites
Tools
Help
Favorites
AV Spring 2011 Beta Product
HR Global Office Calendar
AV Spring 2011 Beta Page
CA Matching Gifts
Global Support Statistics
CA Agile General Discussion ...
CA 2E - Message boards
2E new wiiki
CA 2E r8.6 Release Project l...
Home - Vision Products
Salesforce.com - Custom...
Page
Safety
Tools
salesforce
SOFTWARE
Bookmark this page
User Name
Password
☐ Remember User Name
Login
F your passwo
Su er 11
n, , an oc
Internet
100%

L abo eatures
ar atch the Webinar now

Do ave ccoun ign up f free.

See what Chatter can do for your company

Chatter vide

Buil our bus s r the c

alcula

T oud Computing Ev f the Year

Dre e: save $200

Security Alerts, Best Practice nd Syste atu

To review security alerts, learn more ab rotecting y lest CRM information and access, and revie tem status trust.salesforce.com

Learn More

New Us

Please retr our user name and temporary password from your em ccou r contact your organization's Salesforce CRM trator for further instructions.

https://trust.salesforce.com/

Privacy Note: There is no mission more important to salesforce.com than protecting the security and privacy of your data. If you ever receive a suspicious email requesting your salesforce.com account information, please contact us immediately at abuse@salesforce.com for a prompt investigation and response.

© Copyright 2000-2011 salesforce.com, inc. Customer Relationship Management (CRM) All rights reserved Various trademarks held by their respective owners.

Agile Vision

DEMO

EPICS

Essentially a story that is too big to be completed in one Sprint.

Epic is broken down into separate stories

Grooming the backlog

Modify description/title.
Estimate size of unsized stories.
Prioritize/re-prioritize.
Target/untarget for release.
Add/remove/change acceptance criteria.

Backlog grooming can be done a little at a time.
2E team has a backlog grooming session week 2 or 3 of each sprint.

Estimating size of stories

Hours vs. story points

Poker planning

– Collaborative exercise to give a ‘good enough’ estimate

User stories, story points and planning poker

What is a User Story?

“In agile development, the user story is a lightweight and more nimble substitute for what has been our traditional means of specifying software requirements - software requirements specifications, use case specifications, and the like.”

Source: Leffingwell LLC, 2009

Sample user stories

As a <user> I can <activity> so that <value>.

As an online banking customer **I want to** ensure that only I can access my account **so that** my information is kept private.

As a patient **I can** book an appointment online.

As a video-sharing-site customer **I can** upload funny videos **to make** people laugh.

As a video-sharing-site CEO **I want to** be able to identify and remove copyrighted material uploaded to the site **to prevent** law suits.

“INVEST” in your stories

Independent

Neogitiable

Valuable

Estimable

Small

Testable

Source: http://xp123.com/xplor/xp0308/

Story points

> *"Story points are a unit of measure for expressing the overall size of a user story."*
>
> Mike Cohn, Agile Estimating and Planning

- A Story's story points encompass: Effort and Complexity and Risk.
- No formula is available or used.
- Points <> hours
- The literal value is meaningless.
- The relative value is important: 5-point story is 5 * size of a 1-point story
- Typical values: 1, 2, 3, 5, 8, 13, 21, 40 & 100

Advantages of story points

Studies show that people are better at relative estimating

— *Charlottesville to Ohio vs. Charlottesville to California*

Story points drive cross-functional behaviour

Story points typically faster to estimate than hours/days

How to get started estimating Story Points?

Story points <> hours CHECK!

It’s all about the Relativity CHECK!

How to estimate your first story???

A. Select a small story and assign it a value of 1.

or

B. Select a medium story and assign it a value of 5.

Then compare other stories to these seed stories.

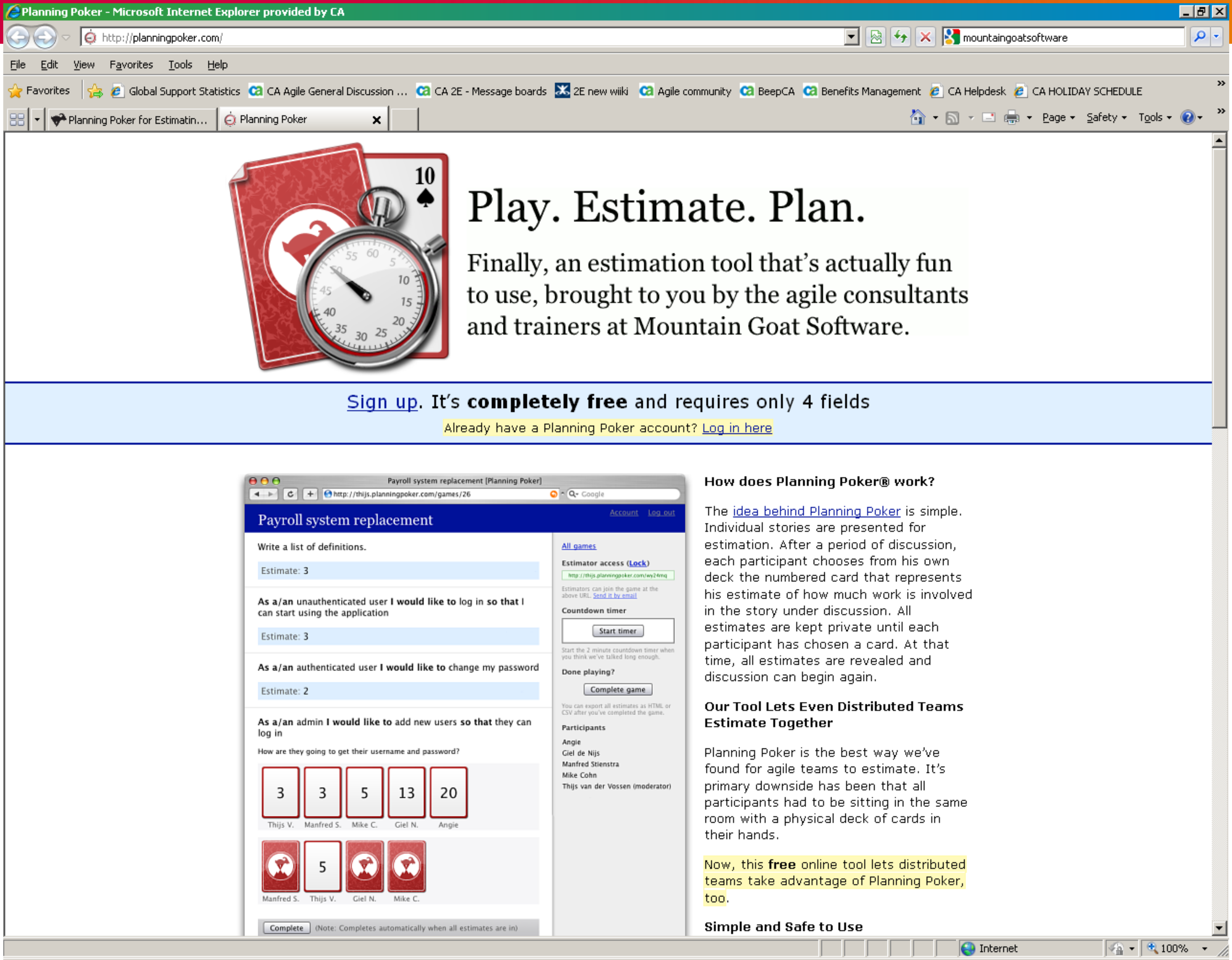
What is planning poker?

- A fun story point estimation technique.
- The team who will implement the stories gather in a room.
- Each has a deck of cards: 1, 2, 3, 5, 8, 13, 21, 40, and 100 and ?.
- A User Story is described by a moderator.
- Each team member estimates the size and places a card face down.
- When all cards are down they are turned over.
- Discussion ensues.
    - Justify estimates, especially outliers
- If required the cards can be gathered and replayed.
- Repeat until a consensus is reached.

Why use planning poker?

- Widely used in the agile community
- Fast and accurate-enough
- Blends expert opinion, analogy, and disaggregation
- Cross-functional input
- All voices are heard
- The dialogue improves the estimate
- Teams find it to be Fun

Planningpoker.com



Play. Estimate. Plan.

Finally, an estimation tool that's actually fun to use, brought to you by the agile consultants and trainers at Mountain Goat Software.

Sign up. It's **completely free** and requires only 4 fields

Already have a Planning Poker account? Log in here

Payroll system replacement [Planning Poker]

http://thijs.planningpoker.com/games/26

Account Log out

Payroll system replacement

Write a list of definitions.

Estimate: 3

As a/an unauthenticated user **I would like to** log in **so that** I can start using the application

Estimate: 3

As a/an authenticated user **I would like to** change my password

Estimate: 2

As a/an admin **I would like to** add new users **so that** they can log in

How are they going to get their username and password?

3	3	5	13	20
Thijs V.	Manfred S.	Mike C.	Giel N.	Angie

	5		
Manfred S.	Thijs V.	Giel N.	Mike C.

Complete (Note: Completes automatically when all estimates are in)

All games

Estimator access (Lock)

http://thijs.planningpoker.com/wy24mq

Estimators can join the game at the above URL. Send it by email

Countdown timer

Start timer

Start the 2 minute countdown timer when you think we've talked long enough.

Done playing?

Complete game

You can export all estimates as HTML or CSV after you've completed the game.

Participants

Angie
Giel de Nijs
Manfred Stienstra
Mike Cohn
Thijs van der Vossen (moderator)

How does Planning Poker® work?

The idea behind Planning Poker is simple. Individual stories are presented for estimation. After a period of discussion, each participant chooses from his own deck the numbered card that represents his estimate of how much work is involved in the story under discussion. All estimates are kept private until each participant has chosen a card. At that time, all estimates are revealed and discussion can begin again.

Our Tool Lets Even Distributed Teams Estimate Together

Planning Poker is the best way we've found for agile teams to estimate. It's primary downside has been that all participants had to be sitting in the same room with a physical deck of cards in their hands.

Now, this **free** online tool lets distributed teams take advantage of Planning Poker, too.

Simple and Safe to Use

Planning Poker demo (http://simoncockayne.planningpoker.com/ht17xl)

Velocity

Velocity = Total Story Points of User Stories completed in a Sprint

Many ways to analyze

- Velocity of previous sprint
- Mean Velocity of last 12 sprints
- Mean Velocity of 5 ‘worst’ in last 12 sprints
- …

No partial credit

Which stories go into the Sprint Backlog? (reprise)

Product Backlog

User story (titles)	Story Points
17929829: Improve ILE Return Code Error Handling in Batch Jobs	8
18180213: 2E RESULT SET SUPPORT	40
18000002: Ability to view the parameters of locked functions	8
…	

Sprint planning

Velocity = 15

Sprint Backlog

User story (titles)	Story Points
17016392: Hidden (obsolete) objects	8
18000001: Add *BUILT IN functions for string handling	5
BUG: 17490083 Non Key Field not Gen CHGOBJ	2

User stories, story points and planning poker Summary

- User stories
  - nimbler substitute to traditional requirements doc.
  - Card, Conversation and Confirmation.
  - INVEST.
- Story points
  - No set formula.
  - Function of effort, complexity & risk.
  - It’s all about the Relativity.
  - Velocity helps plan Sprint (Spring Backlog).
  - Cannot and must not compare velocity between teams
  - Don’t get hung up on story points. Uncomfortable at first.
- Planning poker
  - Supports cross-functional team input.
  - Blends expert opinion, analogy & disaggregation.
- **Most important aspect is the team**

Agenda

Pre-release testing

Agenda

Pre-release testing

CA 2E r8.6

- Customers have to sign pre-release testing NDA.
- End of sprint reviews at end of every sprint.
- We are ‘shipping’ code every four weeks.
    - Not to be used in live/production environment.
    - Not supported as GA product.
- Secure FTP site for download.
- Feedback via Private forum (category on CA community).

Useful resources

Useful external links

Agenda

Books

- “Agile Estimating and Planning”, Mike Cohn
- “Agile Software Development with Scrum”, Ken Schwaber
- “Agile Project Management with Scrum”, Ken Schwaber
- “Adapting Configuration Management for Agile Teams”, Mario E. Moreira

Web

- http://agilemanifesto.org/
- http://www.scrumalliance.org
- http://www.mountaingoatsoftware.com
- http://www.oobeyagroup.com/
- http://planningpoker.com/
- http://innovationgames.com/
- **CA Vision Products Global User community:** https://communities.ca.com/web/agile-vision-global-user-community/message-board/-/message_boards/category/2942583

Summary

“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 on the items on the right,

we value the items on the left more.”

http://www.agilemanifesto.org/

Q&A

Agenda

thank you