EOL-CDS Experiences with Agile SCRUM

20 October 2011

Introductions

Sandra Thurn
thurn@ucar.edu

- *EOL Role:* In Field Project Services (FPS); Project Management process development and technical project management
- *Industries:* Aerospace, Scientific Research/IT
- *Agile Experience:* Managing software development teams at Rockwell-Collins; Project management at NCAR; Lean practitioner
- *Degrees:* BA Math, BSEE, MBA, ME (Master of Engineering, Systems)
- *Certifications:* Certified ScrumMaster (CSM), Project Management Professional (PMP)

Greg Stossmeister
gstoss@ucar.edu

- *EOL Role:* Head of Collaborative Tools and Metadata Group (CTM) in the Computing Data and Software facility (CDS)
- *Areas of Expertise:* Data Management, Field Project Operations
- *Scientific Research:* Tropical and Mesoscale Meteorological Research
- *Software Engineering Experience:* Scientific analysis, Data visualization, File System Management
- *Degrees:* BS Physics, MS Meteorology

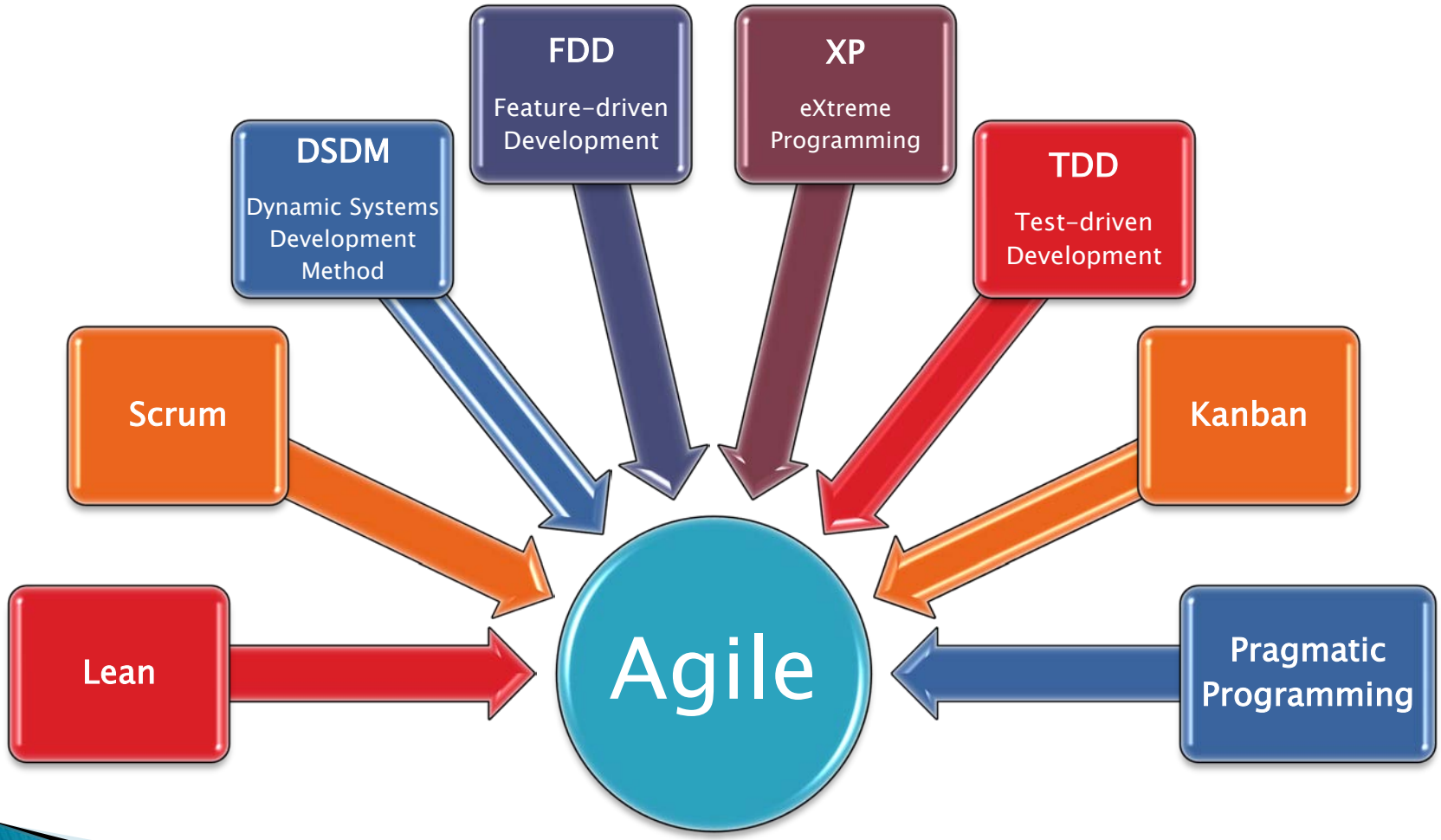
Agenda

- What is Scrum?
- Why did EOL use it?
- Requirements and User Stories
- EOL-CDS Implementation Details
- Summary and Recommendations
- Websites and Reading List

What is Scrum?

- What is Agile?
- What are the elements of Scrum?
- How is it different from traditional methods?

Agile Methodology Family



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

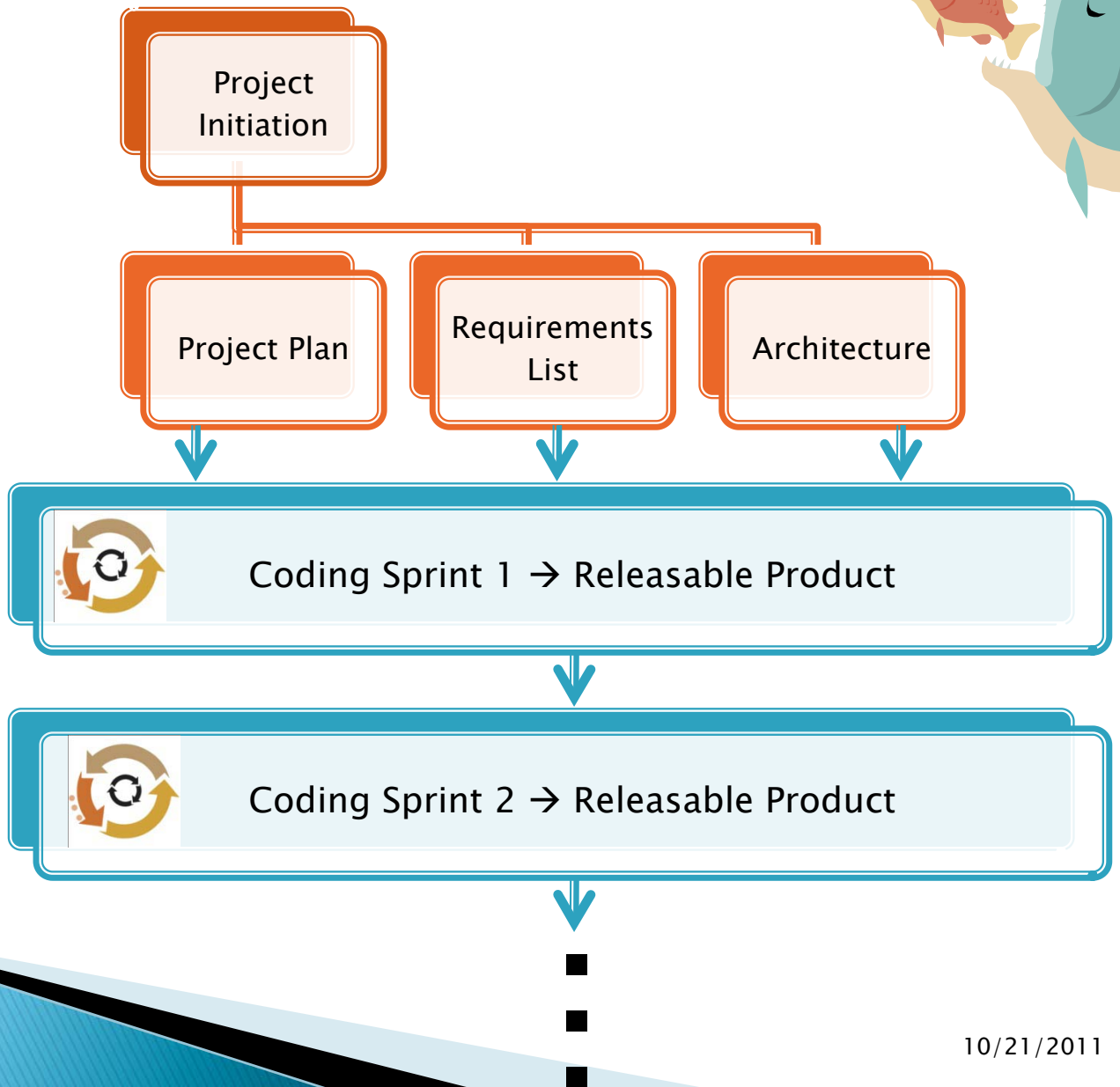
- ***That is, while there is value in the items on the right, we value the items on the left more.***

- **Signed by (Feb 2001):**
  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

Scrum Characteristics

- A **framework** for managing projects
- Self-organizing teams
- Product progresses in a series of “sprints”
- Requirements are captured in a list that is prioritized based on customer value
- No specific engineering practices prescribed
- Uses evolving rules (e.g., team agreements) to create an agile environment for delivering products
- **Scrum is NOT command-and-control**

Agile Life Cycle



Scrum framework

Roles

- Product Owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Daily scrum meeting
- Product demo
- Sprint retrospective

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Roles

- Product owner
  - Define and prioritize product features; adjust every iteration, as needed
  - Provides input and makes decisions daily
  - Accept or reject work results
- ScrumMaster
  - A servant leader role
  - Responsible for enacting Scrum values and practices
  - Remove roadblocks
- Team
  - Cross-functional: Programmers, testers, etc.
  - Membership changes only between sprints
  - Self-organizing

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

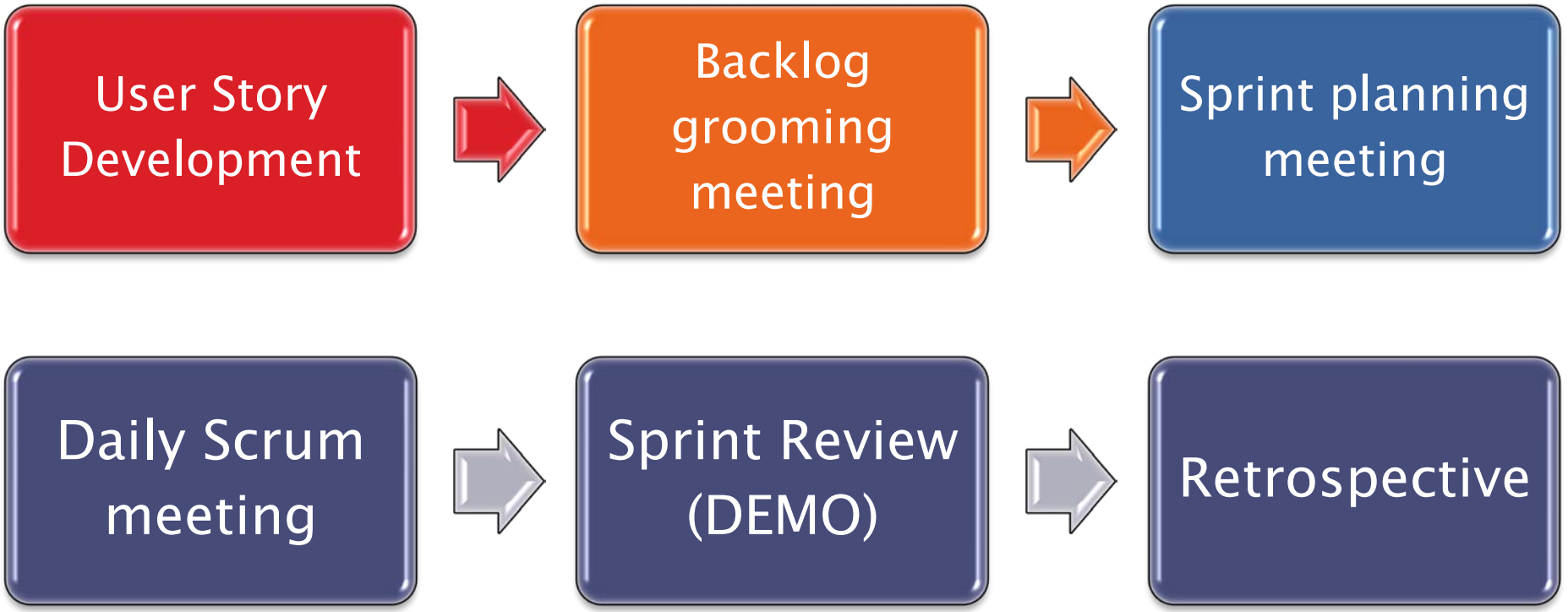
Ceremonies

- Sprint planning
- Daily scrum meeting
- Sprint review
- Sprint retrospective

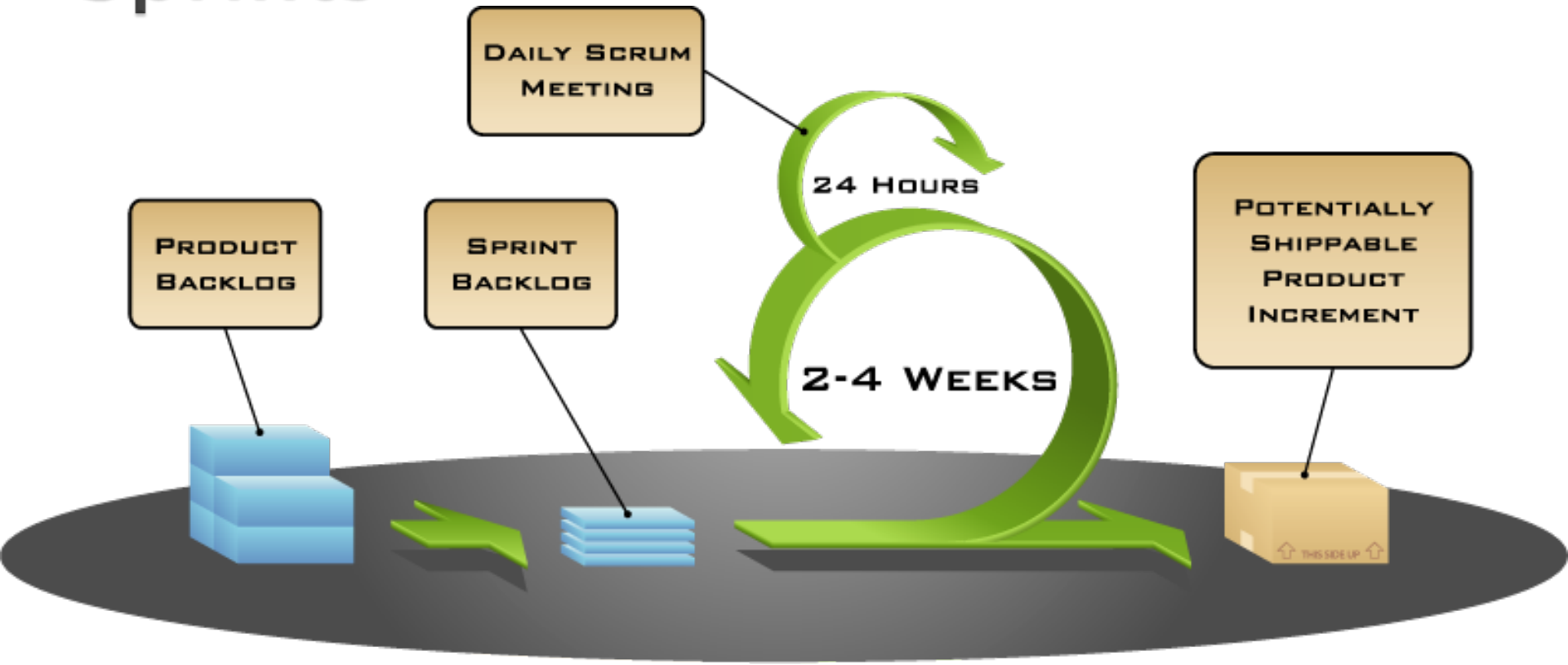
Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Ceremonies


User Story Development
Backlog grooming meeting
Sprint planning meeting
Daily Scrum meeting
Sprint Review (DEMO)
Retrospective

Sprints



User Story Development → Backlog grooming meeting → Sprint planning meeting → Daily Scrum meeting → Sprint Review (DEMO) → Retrospective

The Daily Scrum

- Parameters
  - Daily Stand-up
  - 15-minutes
- Not for problem solving
  - Whole world is invited
  - Only team members, ScrumMaster, product owner, can talk
  - These are *not* status for the ScrumMaster -- they are commitments in front of peers
- Helps avoid other unnecessary meetings

Everyone Answers 3 Questions

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product backlog

- “The requirements”
- List of all desired project work
- Expressed in terms of its value to the users or customers
- Prioritized by the product owner
- Reprioritized at the start of each sprint (Backlog Grooming)

This is the **product backlog**

EOL-CDS SCRUM Experiences

- How was Scrum applied in the EOL Computing Data and Software services facility?

  How did it work out?

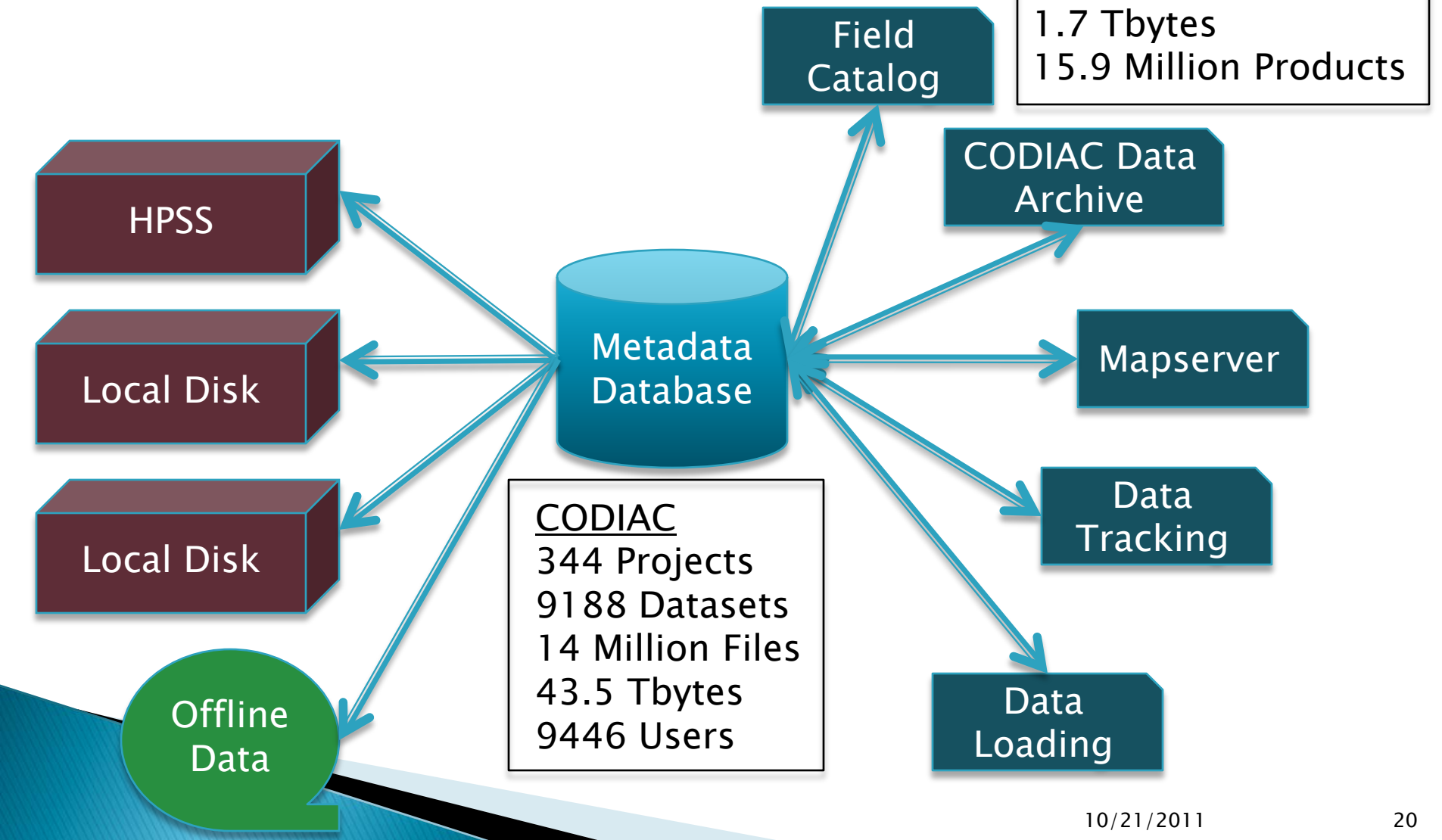
Why Scrum for EOL-CDS?

- Addressing these two problems lead us to try Scrum
- 1. *Software development engineers also are deployed in the field*
  - Generally cannot work with development team while gone
  - Lose continuity and need to review / revise work when they return
- 2. *Software development engineers support multiple areas*
  - Multitasking environment leads to inefficiencies
  - Average phone call interruption takes 5 minutes, re-immersion period is 15 minutes
  - Re-immersion period if away from something for days or weeks can be significant
- **Scrum addresses these issues very well!**
  - Iterations are completed through "release-quality" so re-immersion into the code not needed
  - Completed units can be used in the interim contributing to code maturity
  - Short-term focus increases efficiency and quality (accuracy?) by reducing context-switching

EMDAC

- What is EMDAC?
  What is the Field Catalog?
  How did we use Scrum?

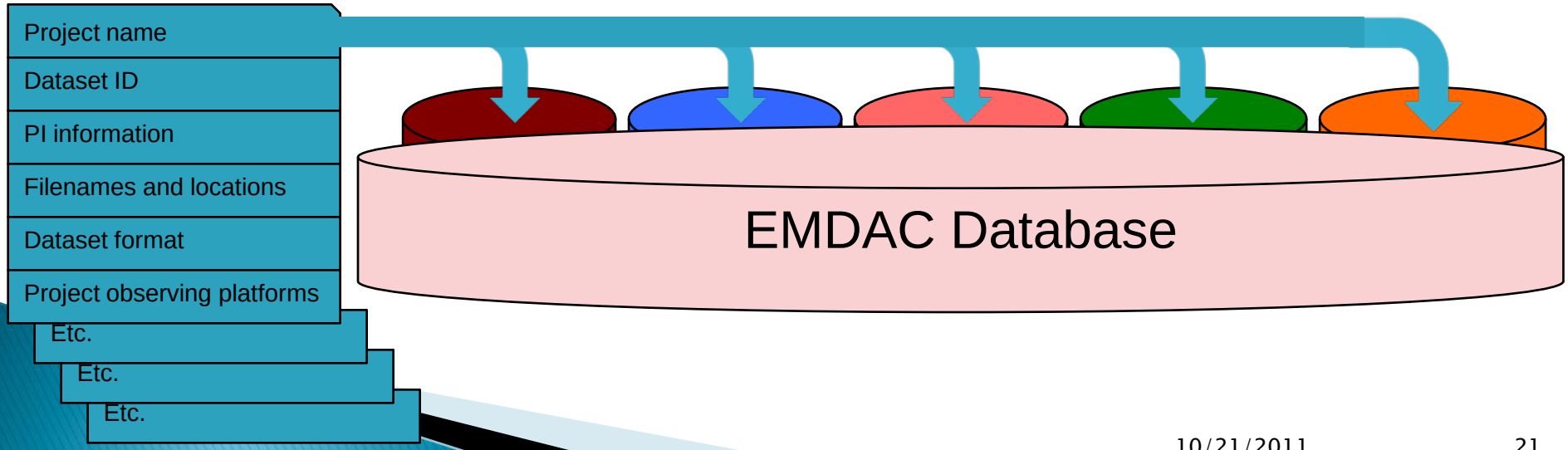
EMDAC – EOL Metadata Database and Cyberinfrastructure



EMDAC Issues/Architecture

- Unify Database Architecture
- Simplify metadata and data access and maintenance
- Allow for new services to be developed for old and new datasets
- Improve Data Access & Data Ingest Interfaces
- Improve Data Discovery Capabilities
- Centralize and Improve Data Metrics Collection and Reporting
- Improve Long-term Data Stewardship

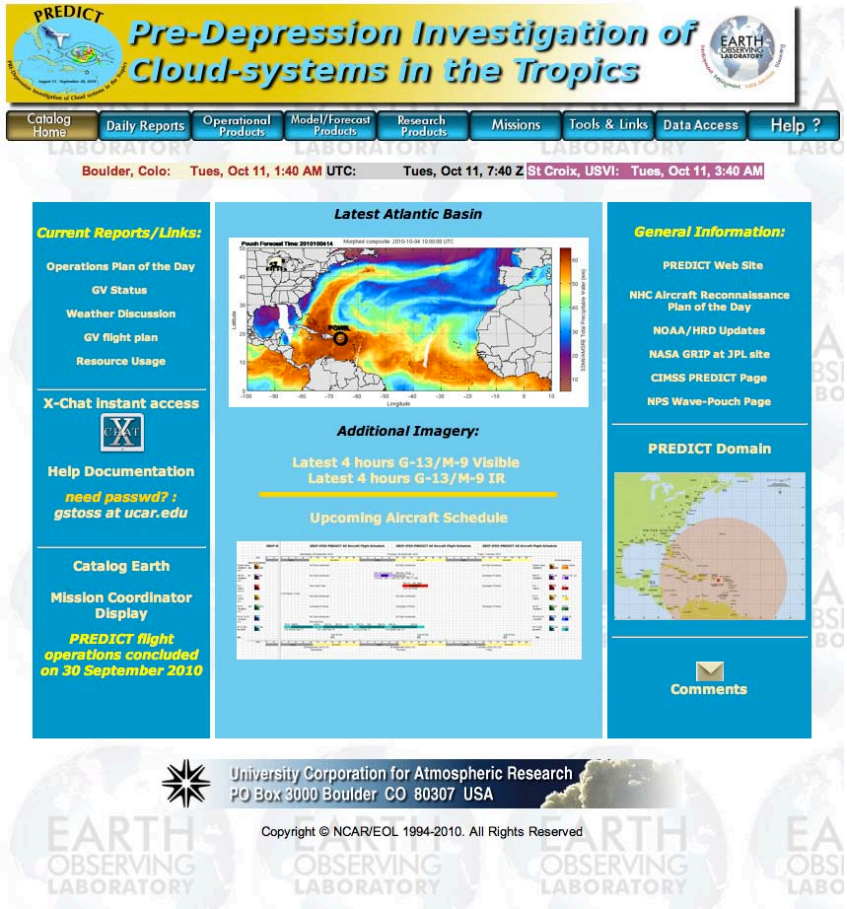
Metadata:



EOL FIELD CATALOG TOOL

In-field tool to ingest and display operational and preliminary research products and project documentation for making real-time decisions and evaluating project progress

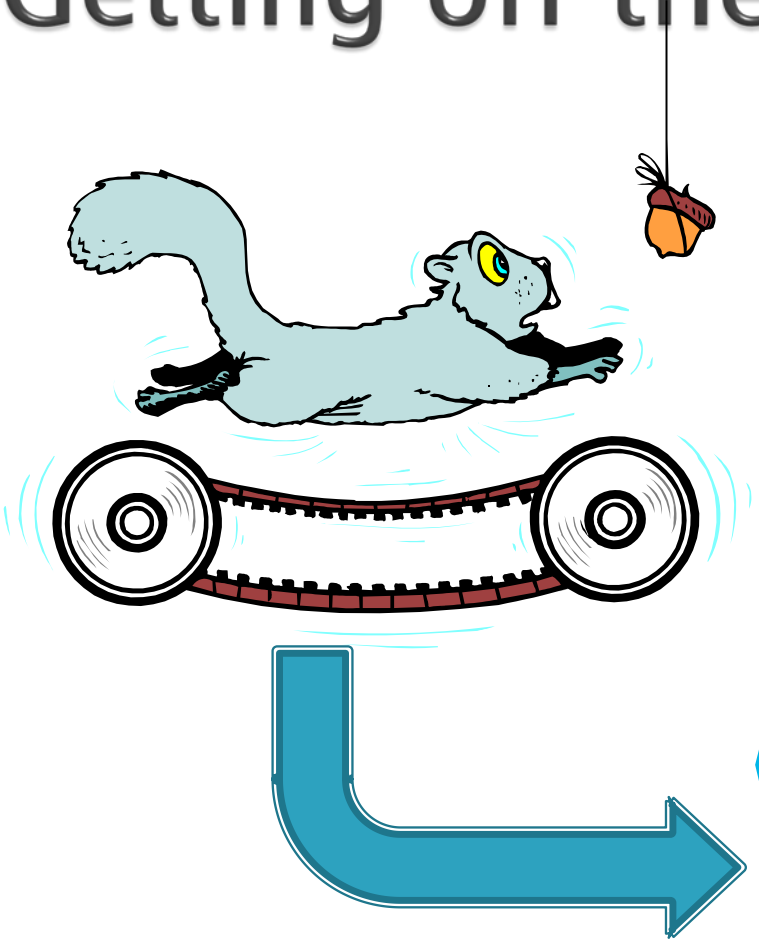
- Daily Mission Reports
- Operations Summary
- Facility Status Reports
- Data Analysis Products
- GIS-based display
- Preliminary Data Sharing
- Authoring Tools
- Web-based access



EMDAC Issues – Field Catalog

- Originally designed in 1995
- Not base funded
- Series of one-off software packages, customized for each campaign
- 60 field projects and counting (5.4/year)
- Product volumes often exceeding 200 Gb/project, > 1 Million files/project
- Need to handle data sharing in the field
- Need to provide GIS overlay capabilities, particularly for playback
- Desired consolidation of EOL field products from various facilities into a unified website

Getting off the treadmill…

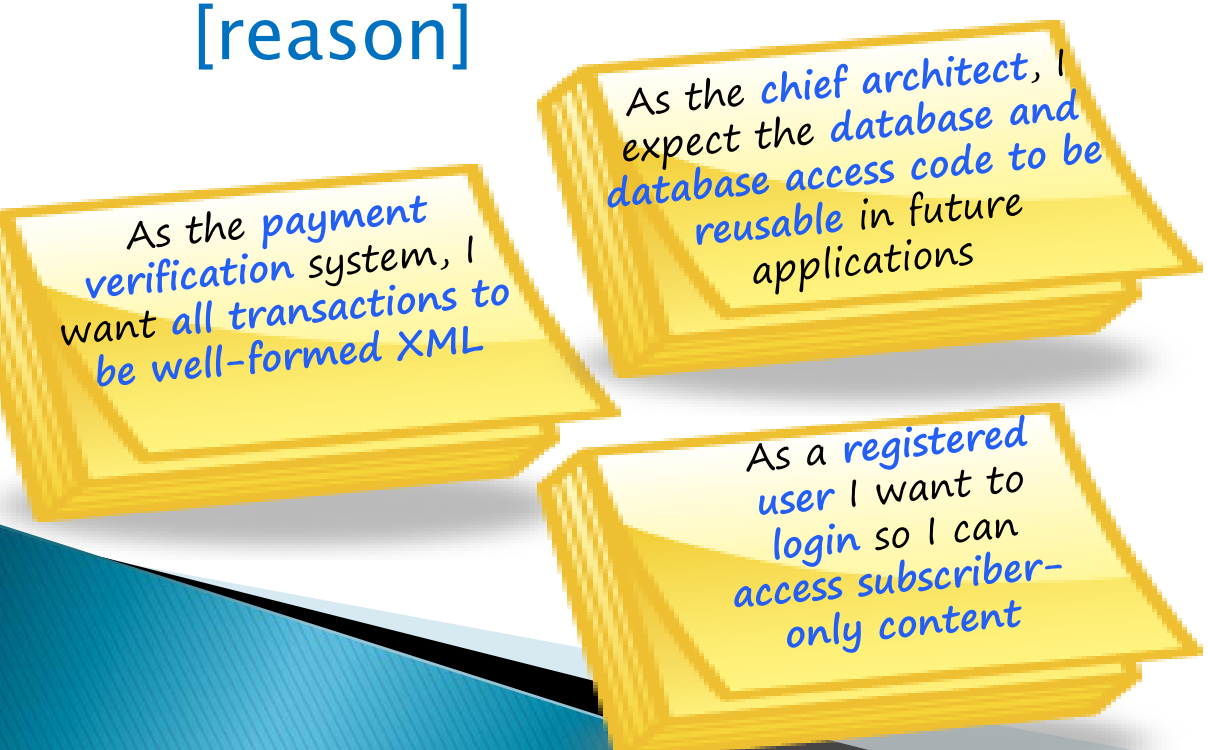


EOL Agile Scrum Implementation Overview

- Requirements and User Stories
- EMDAC – Design Database Schema Sprint
- FIELD CATALOG – Design Catalog Architecture Sprint
- EMDAC – Coding Sprint – Zinc crud application for new schema
- Ongoing – Blend of Scrum and iterative development with one person dedicated to each user story / requirement

User Stories – Expressing Value to User

- Product Backlog (requirements) may be written as User Stories
- As a [user role] I want to [goal] so I can [reason]



- User Stories can be augmented with other techniques

  E.g., Use cases, data dictionaries, scenarios, function cards

Requirements Experiences

- Requirements gathering with users
  - Addressed features with a wide variety of users
  - Considered bug fixes as well as future enhancements based on changing technologies
  - Included requirements for data stewardship, reliability, and flexibility
- User Stories created for Interface Navigation
  - Found we needed to do basic demos to get specific comments, otherwise too abstract
  - New user stories will emerge based on demo feedback
- Note that requirements are fleshed out just-in-time in Agile rather than as a huge upfront effort as is usually done in traditional methods

EMDAC Schema Sprint

Goal

- Develop a draft schema for EMDAC

Tasks:

- Detailed review of existing databases
- Discuss options for consolidation
- Proposed unified database structure
- Document proposed database structure

EMDAC Schema Sprint Highlights

- Developed unified schema that covered 5 separate databases and their applications
- Incorporated schema additions that would allow us to interact with other users and data centers using common metadata standards
- Chose documentation tool that actually created a MySQL database saving time overall

EMDAC Schema Sprint Lessons Learned

These went well…

- Week long focus with breaks so could address other responsibilities
- Participants all had a vested interest
- Participation was success-oriented
- Active and balanced Participation
- Added people when we saw they were needed
- Well-Defined, focused goal
- Having the tasks defined (part of sprint process) so we could recognize progress
- Parking lot helped us keep focused
- ***All Agreed that it went very well and that the positives far outweighed the negatives.***

Next time we will…

- Define sprint expectations better – some expected code sprint style
- Think ahead about the tools and who know how to use them
- One key person not available
- Take notes along the way – ideas were not always recorded.
  Suggestions:
  - Add scribe
  - Action Item entry after each discussion
  - Conclusion or summary written by the group at the end of each topic discussion
  - Group decides on disposition (parking lot, abandon, add to schema) of topic at the end of each discussion AND someone records those decisions

Field Catalog Architecture Sprint

- Goal(s)
  - Develop a field catalog architecture that will support existing features and that will have the flexibility and scalability to facilitate the addition of diverse new features
- Sprint Topics
  - Requirements Review
  - Discuss architectural options
  - Document proposed field catalog architecture
- Sprint Planning
  - Estimate time to devote to each topic
  - Is the right expertise in the room?
  - Decide what form the Sprint Review (demo) will take

Field Catalog Architecture Sprint Highlights

- Developed Block Diagrams of main architectural elements
- Prioritized requirements (32 originally)
- Defined key attributes for each function in block diagrams
- Input considerations
- How to ingest products
- Interactions with data archive
- Ability to view content listings and display products

Field Catalog Sprint Lessons Learned

These went well…

- We did it!!
- Liked having weekend in the middle
- Mix of people – users, developers, scientists, …
- Small group size encouraged input from all participants
- Interactions – energetic and respectful
- Stayed focused
- Omnigraph for diagrams
- Diagrams developed real time

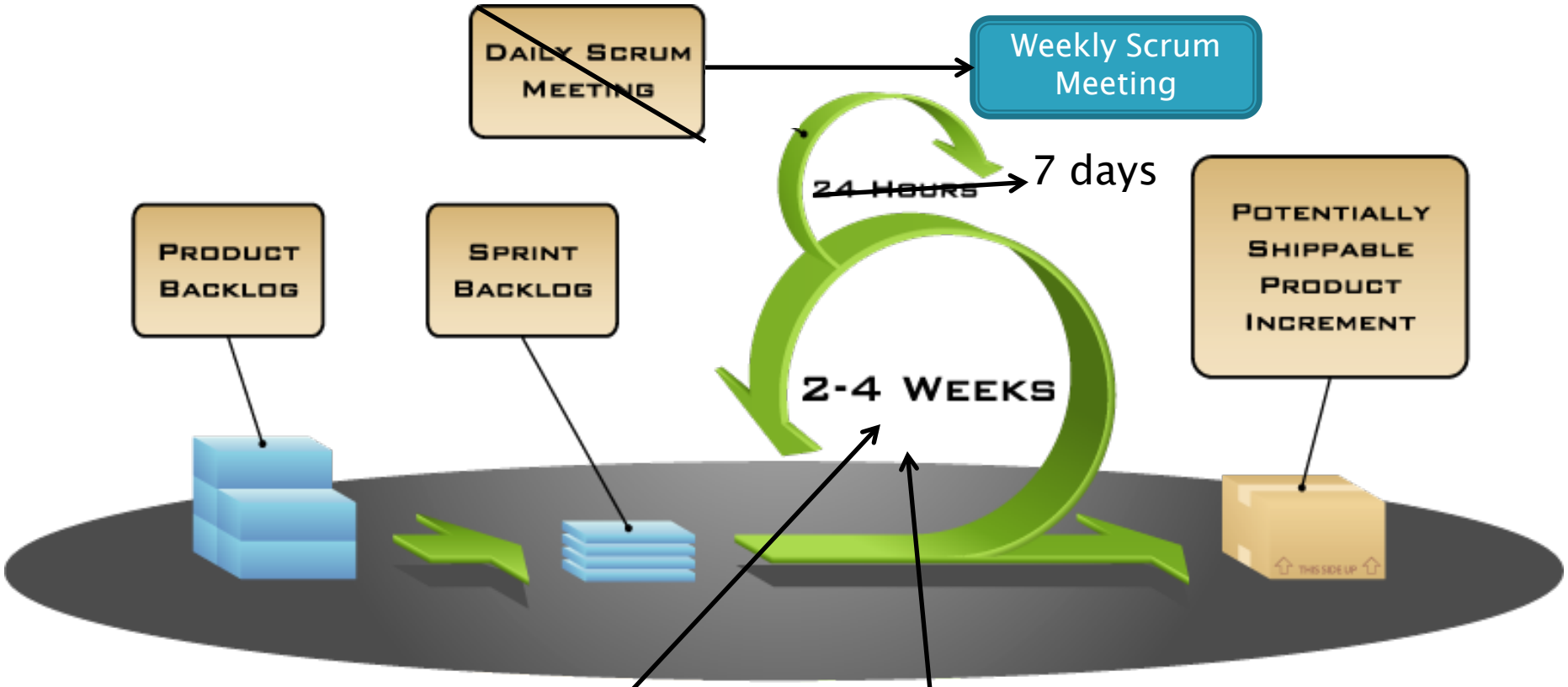
Next time we will…

- Start with overview of EMDAC to establish a common knowledge base
- Definitions up front for some of the basic terminology – e.g., product.
  - Some definitions evolved during the sprint which is a good thing

CRUD Coding Sprint

- *Goal:* Develop replacement application for non-database users to interact with the metadata database and to share knowledge and experience between coders
- Daily Scrum Meetings
- Pair programming
- Sprint Review – working code demo

Ongoing work: EOL Sprint adaptations



Tools

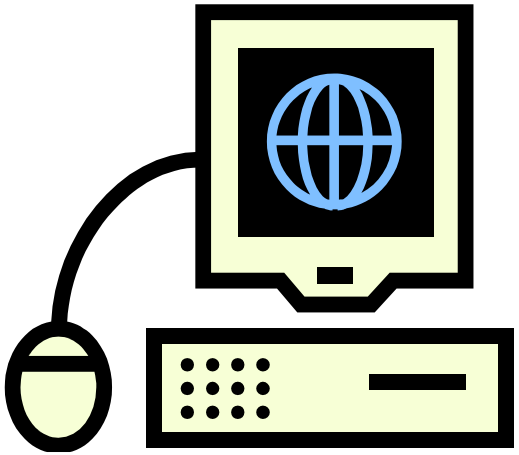
- Using Basecamp (collaborative tool), Visio, and Excel
  - Tried MS Project – not platform-independent
  - Tried Basecamp add-on that had little support
  - Still searching for the best way of tracking progress that flows into higher level progress views (e.g., Gantt charts)
- Examples of Agile tools that could be used
  - Jira with Greenhopper (UCAR has licenses)
  - Rally Software (single project, <10 users free)

EOL Agile Scrum Benefits Summary

- Each work product or feature is usable at end of sprint
- Allows us to adapt to periods of unavailability such as field deployments
- Allows us to adapt to multitasking required by a small and diverse team
- Sense of accomplishment because there is successful completion at every sprint or iteration
- Saves time on requirements because only high level requirements necessary up front

RECOMMENDATION
Be Agile and adapt framework elements for your environment

Where to go next



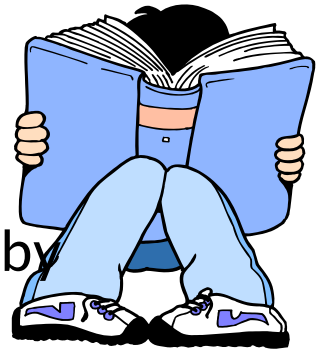
Websites

- www.scrumalliance.org
- www.controlchaos.com
- www.mountaingoatsoftware.com/scrum
- www.implementingscrum.com
- www.balagan.org.uk/work/agile_comparison.htm
- www.rallydev.com
- www.stickyminds.com/s.asp?F=S10365_COL_2 (Article relates PMBOK to Agile)

Local Groups – Meeting Monthly

- Agile Boulder (www.meetup.com/Boulder-Agile/)
- Agile Denver (http://agiledenver.ning.com/)

A Scrum reading list

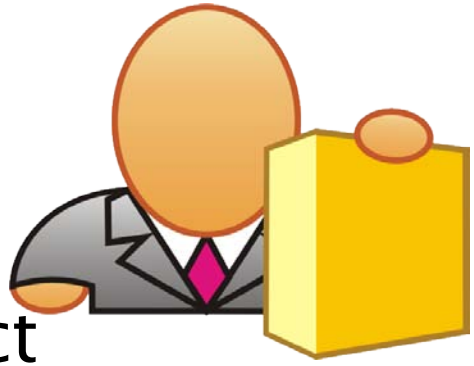


- *Agile and Iterative Development: A Manager's Guide* by Craig Larman
- *Agile Estimating and Planning* by Mike Cohn
- *Agile Project Management with Scrum* by Ken Schwaber
- *Agile Retrospectives* by Esther Derby and Diana Larsen
- *Agile Software Development Ecosystems* by Jim Highsmith
- *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle
- *Becoming Agile in an imperfect World* by Greg Smith and Ahmed Sidky
- *Scrum and The Enterprise* by Ken Schwaber
- *User Stories Applied for Agile Software Development* by Mike Cohn

QUESTIONS?

Supplemental Slides

Product owner



- Define the features of the product
- Decide on release dates and content
- Prioritize features according to value
- Provides input and makes decisions daily
- Adjust features and priority every iteration, as needed
- Accept or reject work results

The ScrumMaster

- A servant leader role
- Responsible for enacting Scrum values and practices
- Remove roadblocks
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Help team manage potential distractions

The Team

- Typically 5-9 people
- Cross-functional: Programmers, testers, user experience designers, QA, etc.
- Membership should change only between sprints
- Teams are self-organizing
  - Ideally, no hierarchy but rarely a possibility

The sprint review (Demo)

- 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
  - No slides
- Whole team participates
- Invite the world

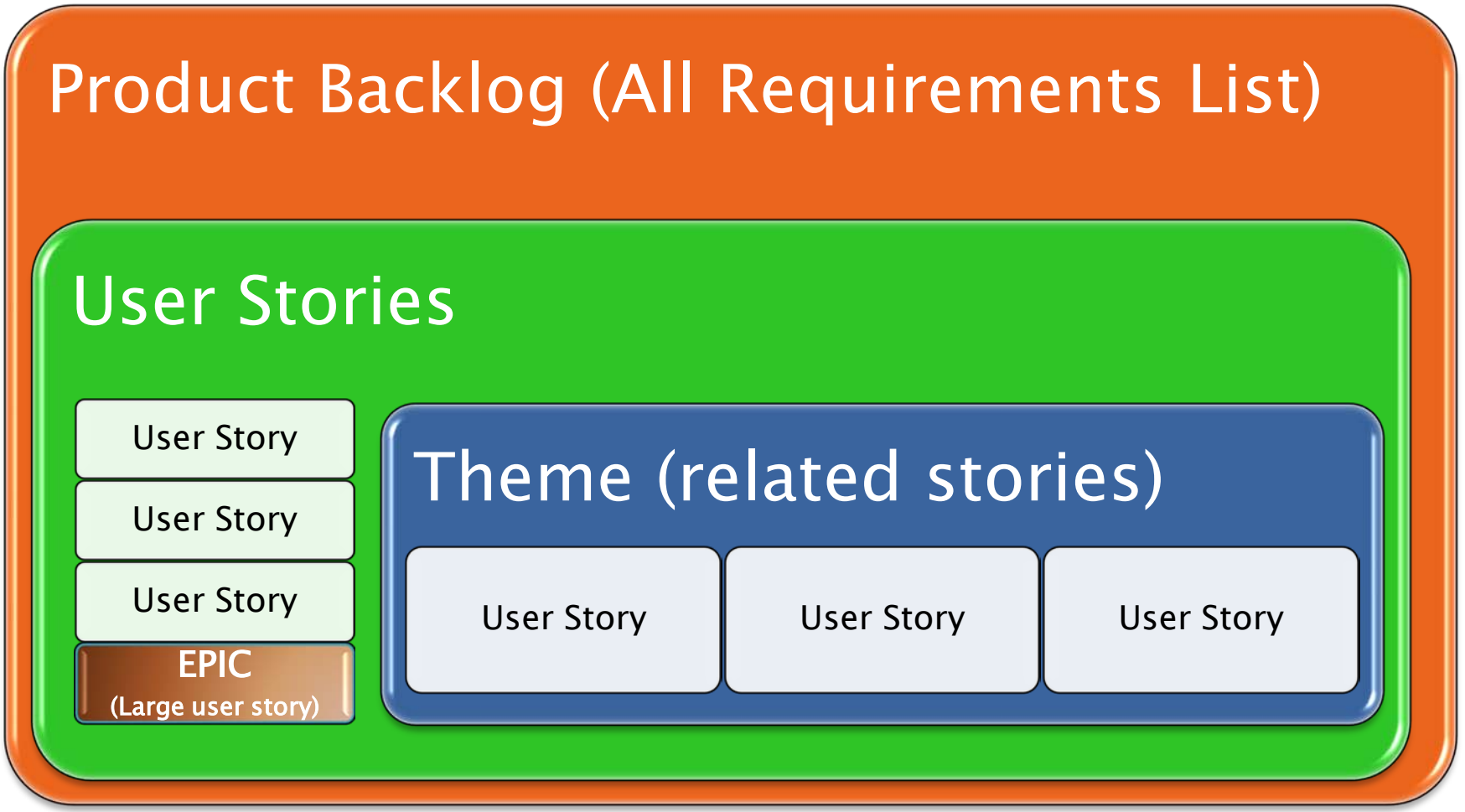
Sprint retrospective

- AKA “Lessons Learned Meeting”
- Periodically take a look at what is and is not working
- Typically 15-30 minutes
- Done after every sprint
- Whole team participates
  - ScrumMaster
  - Product owner
  - Team
  - Possibly customers and others

Keeping Our Sprint Focused

- Ground Rules / Agreements
- Parking Lot / Tough Questions
- Start / finish / break times
- What else?

Stories, Themes, Epics



If story looks complex, use a ***spike***

- First iteration: acquire knowledge
- Second iteration: do the work