THE SOFTWARE PROJECT ROAD KIT

Chris Riesbeck
Two-thirds of software projects run into trouble!

Don't take chances
The Software Project Road Kit

- The Inception Deck
- User scenarios
- Use case diagrams
- MVP

Maps

- Backlogs
- Iterations
- Velocity charts

GPS
INCEPTION DECK

Why are we here?
Better road access?
Logistics tracking?
Regulatory compliance?
Efficiency?

RCS
“The safest way to control traffic on your construction site.”
Instant access - anytime, anywhere
Headache free setup and administration
Fewer accidents
Fewer missed deliveries

Your project community
IS Security     Training
Help desk       Satellite imagery
Infrastructure  Database team

Your core team
Everyone else!
... is always bigger than you think!

Technical architecture
Web application
- C#, MVC.NET
- SQL Server
- NUnit, NAnt, Moq, TeamCity
- ArcGIS (images may be out of date)

Everyone cool with these tools?

Stuff we lose sleep over
- Availability of the customer.
- Not having a dedicated dev/test environment.
- Not have a co-located team.
- Not having full time team members.
- Having ten analysts and two developers.
- Betting the farm on a technology that we have no in-house expertise around.

From http://agilewarrior.wordpress.com/2010/11/06/the-agile-inception-deck/
**What**

- Key questions to answer and share at the beginning of any project

**Why**

- Identify likely show-stoppers
- Get all involved on the same page

**Usage**

- Answer all questions
- Work with developers on technical architecture issues

**When**

- Shares answers with all team members and stakeholders at project kickoff
It's been a hectic day, but Marge finally has 20 minutes for her daily workout. She opens WorkIt! on her iPhone, moves to the center of her living room, puts in her ear buds, presses the big "My Daily 20" button, and pockets the phone in her sweatpants. A few seconds later, she hears the app call out over a disco beat "Warmup time! Ready? Feet apart! Hands on hips! Back straight! Exhale and bend your knees...." About 5 minutes in, the phone rings. "WorkIt pause!" Marge calls out, and WorkIt! pauses...

For more:
http://www.stickyminds.com/s.asp?F=S10730_COL_2
**User Scenarios**

**What**
- Stories of target users interacting with your product in typical contexts

**Why**
- Exposes constraints and key needs
- Basis for pitch demo
- Guides implementation choices

**Usage**
- Create personae for target users
- Tell a whole story, from motivating event to conclusion

**When**
- Develop when defining initial business value proposition
USE CASE DIAGRAMS

UCDs were designed for use cases, which are neither stories nor scenarios, but the diagrams work as well or even better for scenarios.

From http://www.agilemodeling.com/artifacts/useCaseDiagram.htm
**Use Case Diagrams**

**What**
- Overview sketch of
  - all users of the product, including external systems
  - major scenarios for each user

**Why**
- Exposes all users and scenarios that need to be considered
- Guides release planning

**Usage**
- First, draw your users and external systems
- Then, add main scenarios for every user

**When**
- Develop after initial high-value user scenarios nailed down
WorkIt!

Your personal trainer

hands-free coaching

instant diary

100's of downloadable routines

Push a button and go!

For more: http://www.kartar.net/2011/10/when-MVPs-go-wrong/
Minimum Viable Product

What
- Minimum: you can't make it any smaller
- Viable: it has a distinct value
- Product: it's usable and useful

Why
- Get to testing as fast as possible
- No distractions in tests
- Easier to change after tests

Usage
- Be ruthless
  - Ries: cut in ½ then again then again
- Ideal: the one-button app

When
- Design for the most critical user scenario(s)
Minimum Viable Product

I'll know what's in my MVP after I talk with my developers.

Why does this not make sense?
From http://www.mountaingoatsoftware.com/scrum/task-boards
**Prioritized lists of user stories:**
- Release backlog: the stories needed for a release
- Scenario backlog: the stories needed for a user scenario
- Iteration backlog: the stories selected for the current iteration

**Highest valued stories done first**
**No guessing about when stories done**

**Prioritize stories in release**
**Developers pull stories from iteration backlog in priority order**

**From MVP, make release and scenario backlog**
**Create iteration backlog at start of each iteration**
From http://www.planbox.com/features/product-management
**What**
- Fixed timeboxes
  - Deadline never slips
- 1 or 2 weeks (1 better)
- Same size throughout project

**Why**
- Agility
  - Stories can be added and/or reprioritized every iteration
- Provides a standard baseline for velocity

**How**
- Select top-value stories
- Developers tell you when to stop
- Developers return stories not completed to release backlog

**When**
- Throughout implementation
**Velocity Charts**

- Burndown chart graphs story points remaining in release backlog
- Shows over/under shooting of deadline

- Burnup chart graphs story points in release (done and undone) and points completed
- Shows velocity and scope change
**What**

- Simple charts showing story points done and remaining in release backlog

**Why**

- Clearly displays most likely time to finish
- Triggers early necessary rescoping

**How**

- Developers assign effort points to all stories in release backlog
- Developers score points for story when all tests pass

**When**

- Throughout implementation