

Agile Envisioning: Building It Right, The Third Time

Doug Talbott
Bedarra Research Labs
doug@bedarra.com

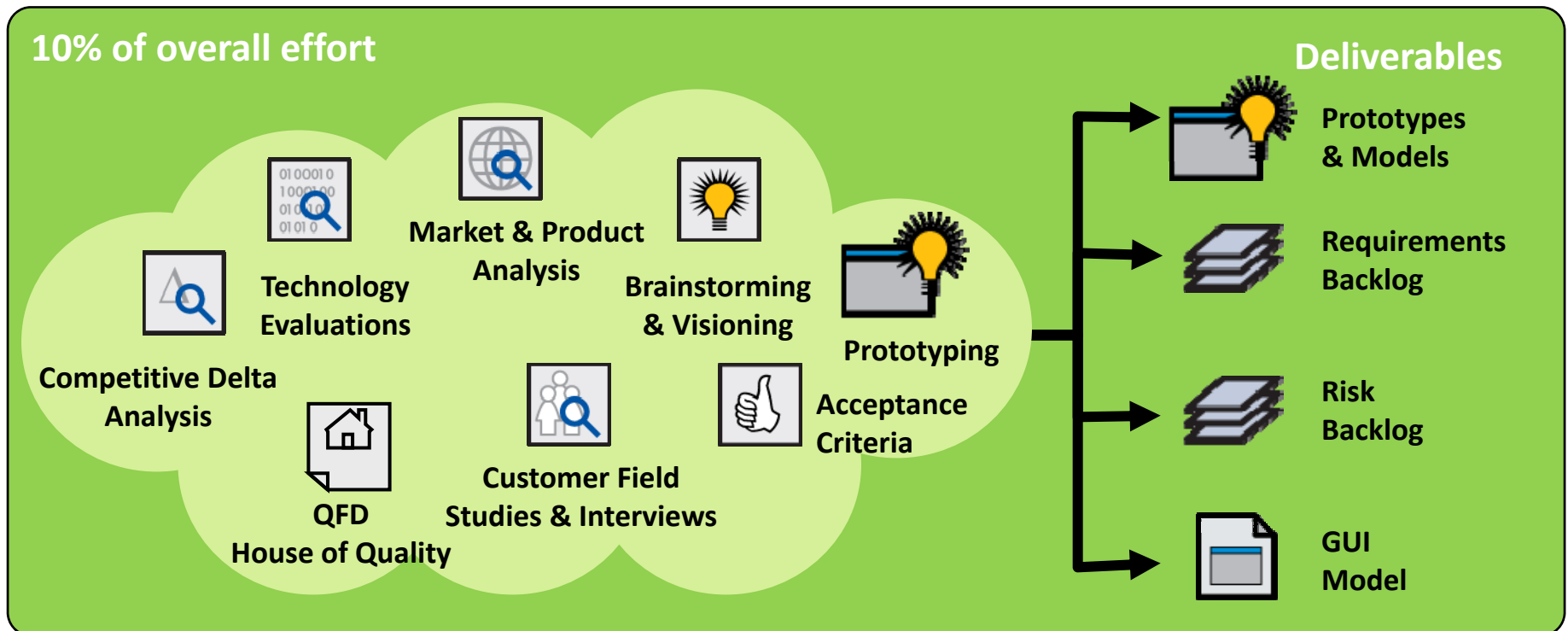


What we're going to cover

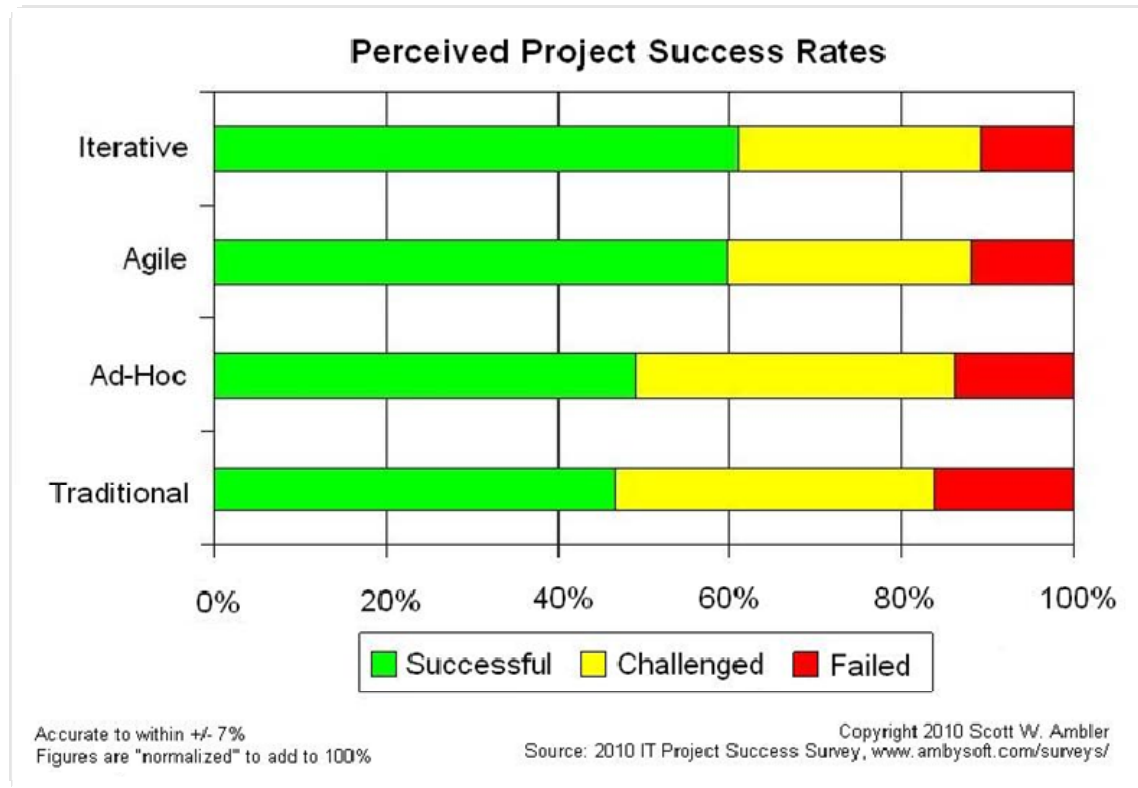
- What envisioning is (definition)
- Why it we need to do it (problem and value)
- When you do it
- Who does it (team, composition, skills, attributes)
- The three main activities
- High-level understanding of the tasks performed
- High-level understanding of the artifacts produced

Envisioning-At-A-Glance

- A light-weight, iterative design activity that couples user-centered design with technology exploration to produce a product vision and roadmap.
- Ensures you deliver right product to right market using right technology through collaboration with end-users
- Practical way of delivering customer-driven design within an Agile process

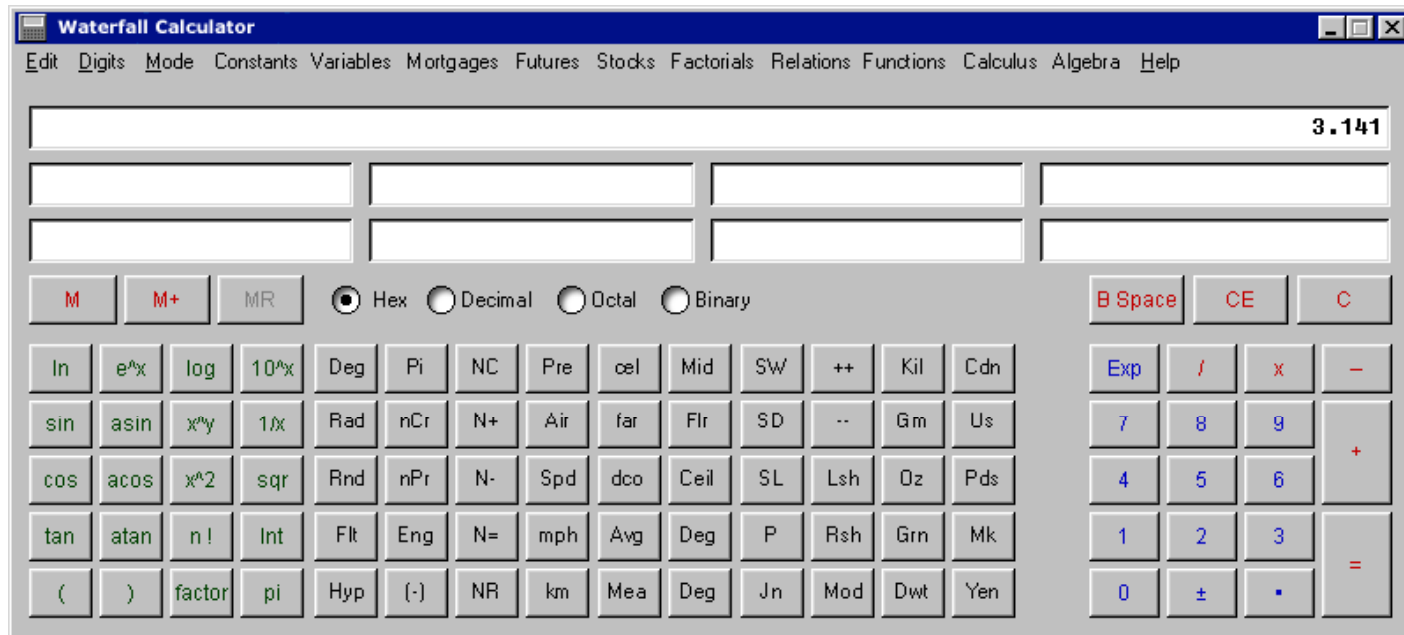


Why Envision?



Why are so many projects still failing?

The Waterfall Problem



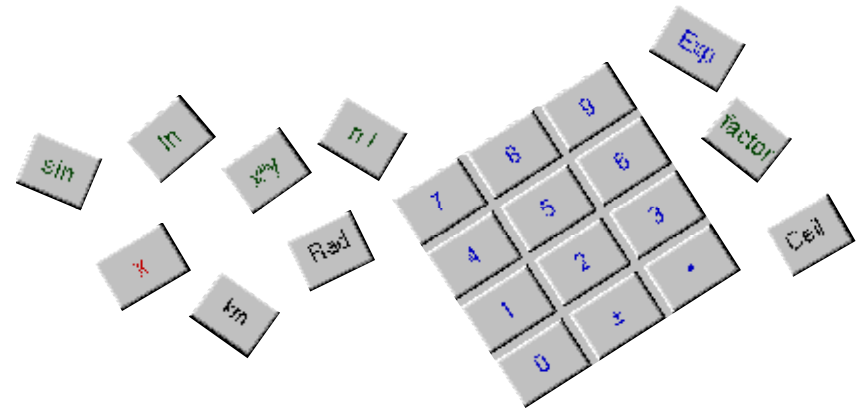
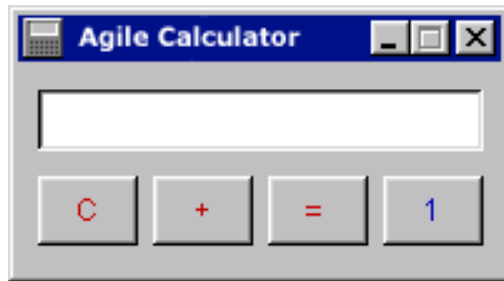
Backlog Symptoms

Huge backlog of stale stories

Gigantic stories

Scope creep

The Agile Problem



Backlog Symptoms

Vague or abstract stories

Lumpy stories in terms of size, clarity, or estimate

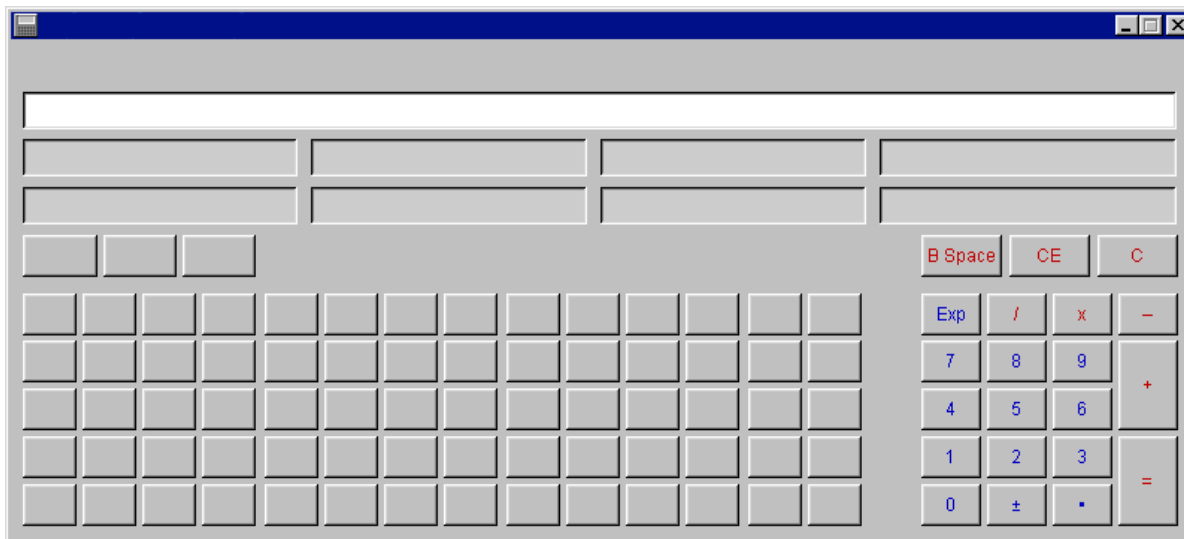
Stories without acceptance tests because we really don't know what the user wants

Lots of new stories getting added to backlog due to missing functionality

Stale stories because we're too busy more important stories

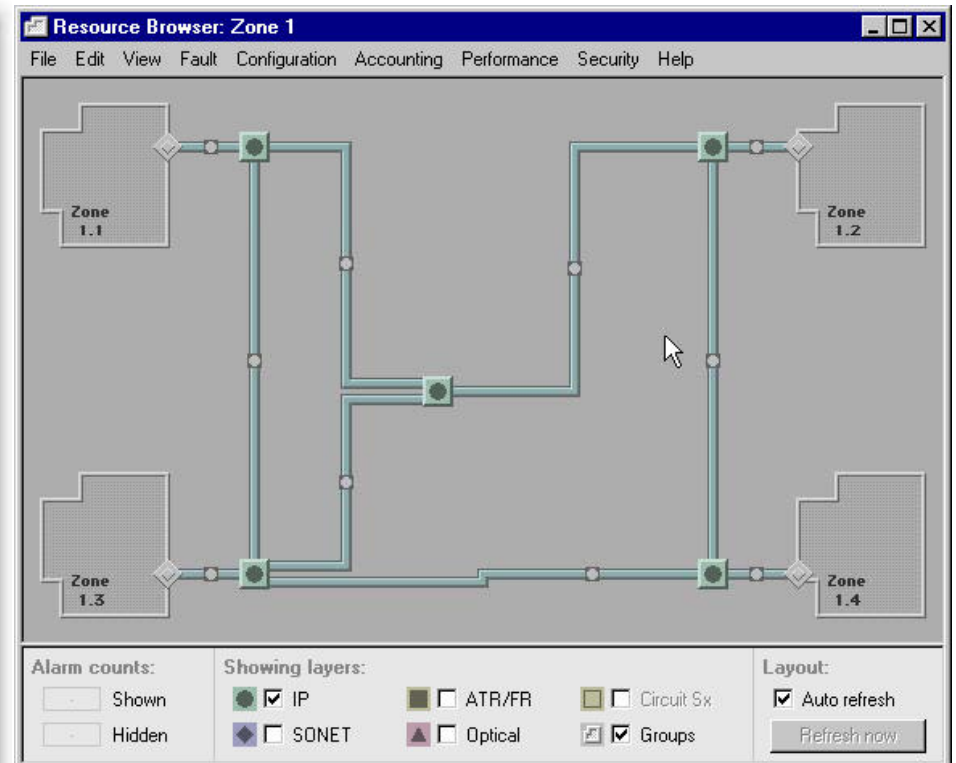
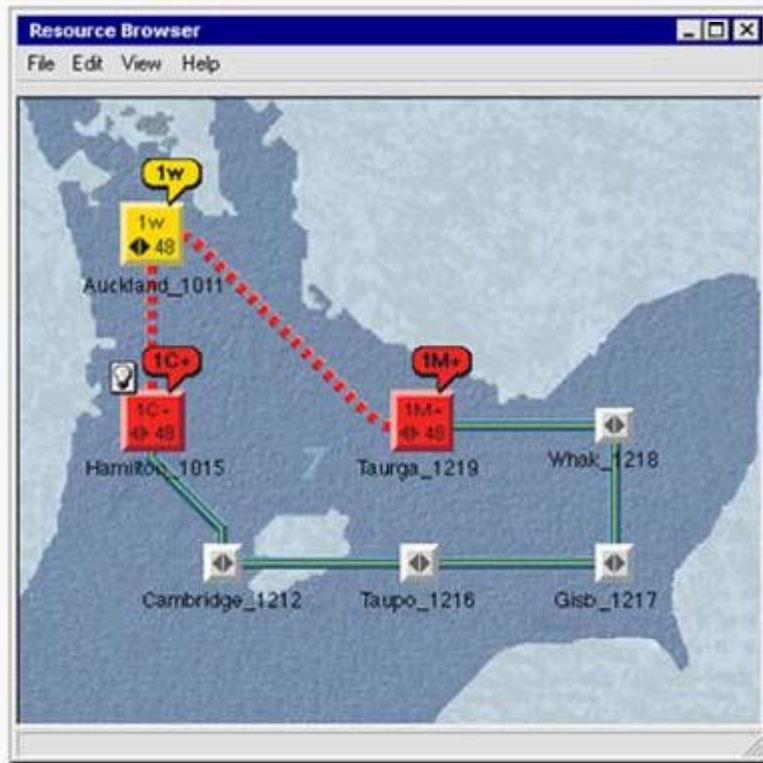
Envisioning strikes a balance

- Understand enough about 'tomorrow'...
- So that you can deliver some incremental value 'today'...
- Without 'developing' yourself into a corner...



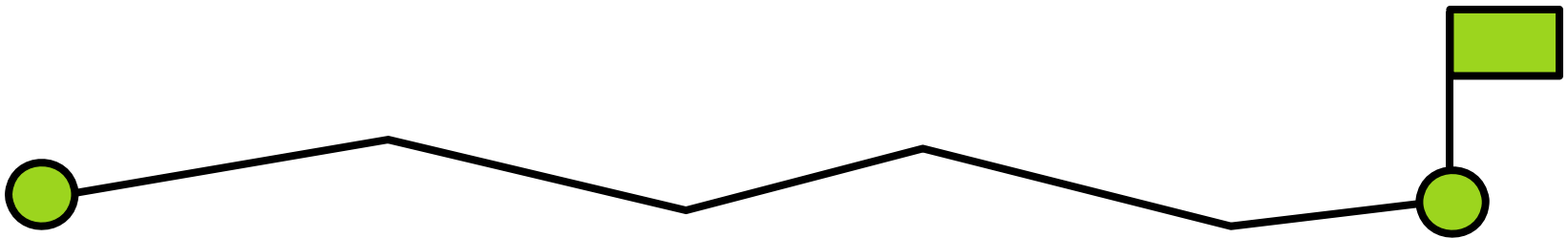
Envisioning Example

Integrated Network Management



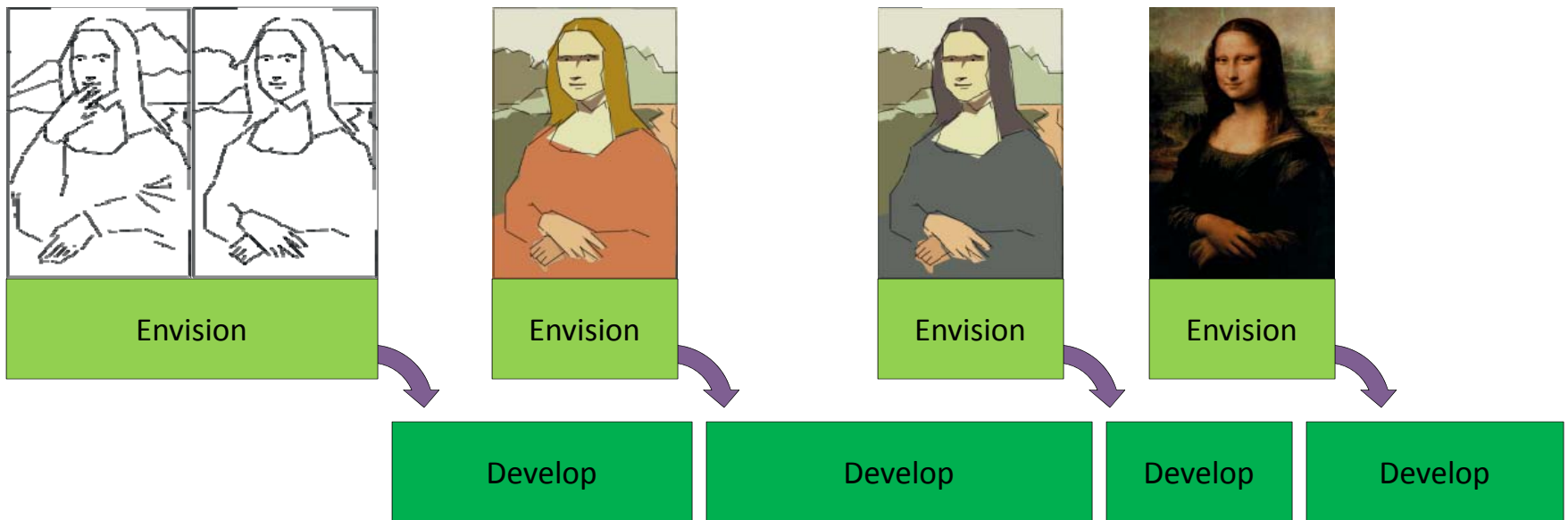
The Value

- Explore product alternatives (value, usefulness, usability)
- Evaluate technology alternatives (feasibility)
- Prototype solutions that make requirements tangible
- Establish acceptance criteria
- Validate those solutions with customers
- Prioritize development based on customer priorities
- Establish a clear product vision and roadmap



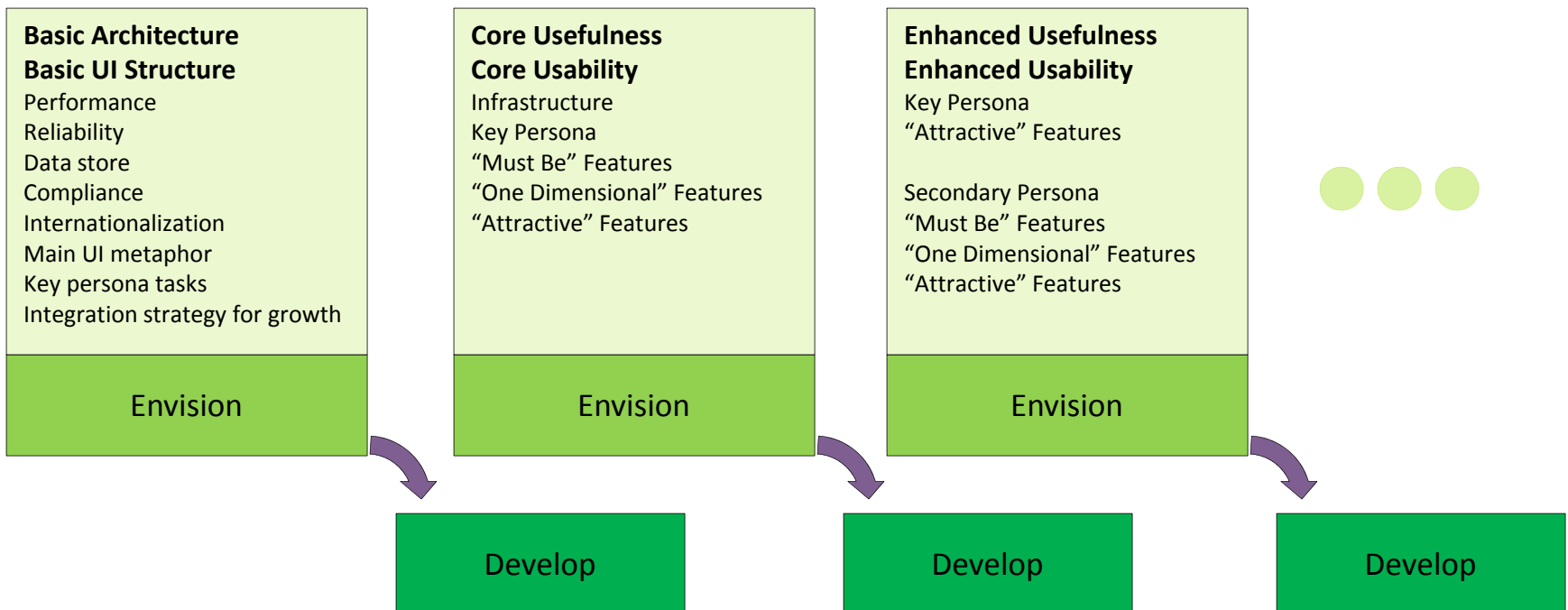
When do you envision?

- ‘Sprint 0’ or before AND as necessary throughout the project...
 - “Sprint 0 has become a phrase misused to describe the planning that occurs prior to the first sprint” *Ken Schawber, co-creator of Scrum*
- When you don’t know how to build something, you should envision it



When do you Envision?

- Important to get the structure right first
- Fixing it can mean lots of work later on



Who Envisions?

- **Multi-Disciplinary Team**

- Business Analyst (act as voice of customer)
- User Experience Designers and Prototyper (paper or interactive prototypes)
- Software Architects, Designers and Prototyper (technical prototypes)
- User Needs Analysts (plan, conduct, report on verification)
- Lead Customer (ideally)

- **Core Skills**

- Thinking (critical, visual, system, lateral, pattern, methodical)
- Expertise (domain knowledge, technical knowledge or skill)
- Collaboration

- **Organization**

- Virtual or physical team room is ideal, co-location is ideal, but not necessary
- SCRUM practices can be applied
- SPRINT practices can be applied

Envisioning Activities At A Glance

Requirements Gathering

1. Identify business value, problems, personas, tasks and task flows
2. Create scenarios and stories

Design Exploration and Prototyping

1. Cluster, categorize and prioritize stories
2. Explore and design architecture
3. Explore and design user interface (as necessary)
4. Software prototype to verify technical feasibility (as necessary)
5. GUI prototype to visualize end-user experience (as necessary)

Design Validation

1. Confirm usefulness and/or usability
2. Confirm form
3. Confirm feature priorities

Requirements Gathering

Purpose, Activities and Artifacts

Purpose

- Identify key value (typically form or function)
- Identify potential opportunities for differentiation
- Identify the users and choosers
- Identify the problem(s) you are trying to solve
- Identify the usefulness and usability needs
in terms of function, form, security, conformance, platform...

**Business Value
or Differentiation**

Problems

**Personas
Users**

**Stories
Tasks**

**Scenarios
Task Flows**

Data Gathering Techniques


- Secondary research - market & competitive analysis (\$)
- Domain expert or craftsman expert evaluations (\$)
- Surveys and questionnaires (\$\$)
- Field studies or job shadowing (\$\$\$)
- One-on-one interviews (\$\$\$)
- Focus group interviews (\$\$\$\$)

Task and Task Flow Analysis Technique

- Why is task analysis so important?
 - **Tasks and task flows** translate nicely to **stories and collections of stories**
- What to do?
 1. Identify and recruit personas
 2. Observe them or talk to them (interview, observation, interview)
 3. Capture the results
- What do you want to learn?
 1. Tasks: What tasks? Priority? Frequency? Issues?
 2. Task Flows: The order in which they perform the tasks?
 3. Success Rates: Percentage of people who succeed at the task.
 4. Completion Times: How long it took them.
 5. Failure Rates: Percentage of people who fail to perform the task.
 6. Failure Points: Why they failed? How it could be improved?

Problem Artifact

- Understand the problem *before* designing the solution
- Understand your “great idea” in the context of solving a “real problem”

Problem	
Name	Cable management problem
Description	Jane has spent hours redecorating her home office, choosing just the right paint & furniture. Her new office looks sleek and modern except for.. the mess of computer & monitor wires and cables snaking around and under her desk. She thinks it looks ugly and distracts from the overall cool new look of the office
Evidence	
Persona Affected	Residential consumer
Severity	Minor

Persona Artifact

- Popularized by Alan Cooper (www.cooper.com)
- Tangible way of representing and relating to the user
- It really hard to write stories for people you can't relate to

Persona	
Name	The persona name (e.g. George Smith) and a type (e.g. Business Consumer)
Gender	Demographic information
Age	Demographic information
Education	Education level
Handicaps	Special issues (e.g. glasses, hearing, physical or mental disabilities, etc)
Technology literacy	Experience with technology, experience with this kind of application or other applications
Cultural bias	Localization issues
Goal and Motivations	Describe the behavioral goal of why the user would want to use this product (e.g. likes to help people, scared of being laid off)
Job Description	Describe the personas role in terms of responsibilities and typical day (i.e. provide the context for using the application)

©2005 <http://personas.quarry.com/docs/QuarrySamplePersonas.pdf>

Partial Persona Example

Jim Robinson | Facilities Manager



“At the end of the day, it’s the facilities manager who’s hung out to dry when things go wrong.”

Age: 46

Education: Electrical Technology Apprenticeship Program at Michigan State University (East Lansing, MI)

Employer: Providence Corporation in Boston, MA

Manages: 35 buildings in MA

Make good things happen for his tenants. Jim works hard to keep his tenants happy, so he wants to know exactly what’s going on in his buildings and how the occupants might be affected. Jim strives to earn high marks on his annual satisfaction survey, as he’s rewarded with a substantial performance bonus.

Role Description

Jim is responsible for all aspects of running 34 of Providence Corporation’s buildings, a mix of field and service offices in Massachusetts. Providence owns all its buildings and rents extra space to third-party tenants in many of them; it’s mostly office space, with a small mix of retail storefronts and restaurants. Jim is based in Boston, in a 200-thousand-square-foot building where he has a small office on the second floor. He is supported by a Customer Service Representative (CSR) in the call center and two technicians who are on the road servicing buildings in his region.

Jim oversees the planning and delivery of services such as cleaning, security, repairs, and small renovations. He is the prime contact for tenant requests in his Boston building, which he receives either in-person or by phone and enters into Providence’s PC-Help software himself. For his 33 other buildings, Jim’s CSR in the call center dispatches work orders to his technicians or, if they’re not available, to a local vendor — but he’ll still get a call if the request is at all unusual or urgent, or if those people can’t be reached immediately. Jim figures he handles about 80 tenant requests on an average day. On those rare occasions when Jim takes a brief vacation, he forwards his office phone to the call center and physically gives his CSR his cell phone so she can take his calls.

Jim is responsible for forecasting the annual cost of managing all 34 of his buildings and for meeting that forecast throughout the year. At the end of each month, Jim checks his performance against the forecast, as he’s evaluated on whether he meets the numbers. Every day, Jim logs into Oracle to approve invoices entered by his CSR, but he also keeps track of his commitments in an Excel spreadsheet. He created this spreadsheet because he knows that neither PC-Help nor Oracle has up-to-date information, which he needs for determining whether he’s met his targets and if he should adjust next month’s forecast.

Avoid a major crisis. Although Jim enjoys his job very much, he has terrible days on occasion. “You’ll feel like toast by the time you get home. You’ll have a headache and will dream about the job all night.” For Jim, the possibility of a major problem or a confrontation with an angry tenant is always just around the next corner.

Jim’s Day

Jim’s day starts by 7:00 am, when he reaches his office in Boston. He grabs his second coffee of the morning before sitting at his desk and plugging in his laptop to check email. “I’m always in Outlook,” Jim explains. “I usually have about 50 unanswered emails to deal with at any given time,” many of which are information requests from his manager or the finance folks at Providence. He also receives email from his tenants and occupants, who are always looking for updates and information about their retail and office spaces. Keeping everyone informed and staying on top of paperwork is a real challenge for Jim. “I try to squeeze it in between phone calls and meetings, but it’s hard to keep up. I’m fighting fires most of the day, so everything else takes a back seat.” He often catches up with paperwork in the evenings at his home office, where he has a high-speed Internet connection.

Jim finds himself on the road about 2 days per week. He avoids overnight trips if possible, choosing to return home each day to spend time with his family even though he doesn’t arrive until late in the evening. He likes to be onsite to personally check on his buildings, especially when major work is underway; but when his workload prevents him from hitting the road, he usually asks his technicians to email digital photos instead. Face-to-face time with tenants is especially critical to Jim, as it allows him to deal with complaints that might otherwise surface only at year-end, on his satisfaction surveys. It also helps him to better understand what people really need vs. what they want. Understanding this allows him to keep them happy while still staying within his budget. Being on the road isn’t easy. He leaves his laptop at home because it’s unlikely he can find a place to dial in during the day. “But this means I return home to dozens of emails. I never go more than three days on the road, because it’s impossible to catch up afterwards. And the call center freaks out when they can’t reach me.” On days when he’s in Boston, Jim gets home by 7:00 pm and spends some time with his wife and two

>>

©2005 RealSuite Inc. and Quarry Integrated Communications Inc. All rights reserved.

Story Artifact

- Standard Format

- As a <user> I want to <do something> so that <benefit>
- Done when <acceptance criteria>

Search Reports
As an analyst, I want to be able to search existing reports in the company report databases so that I can quickly find detailed information to build presentations.
High: N
Low: N

Done when
1. Analyst can search reports by report name (wildcards), abstract contents (wildcards), author name (wildcards), creation, modification, approval dates (yyyy-mm-dd hh:mm) keyword strings, content strings (wildcard, regex), report type (BUS,RES,CONS) and document type (word, excel, ppt, gif, png, jpeg).
2. Returns a list of report with filtering capability on same attributes described in item #1.

- Likely needs ‘envisioning’ before being ‘developed’

- What does the GUI look like?
- How does it fit with the architecture?
- How fast does the search have to be?
- Is this two weeks work?
- Right now, estimating would be a guess at the level of GO-NO GO

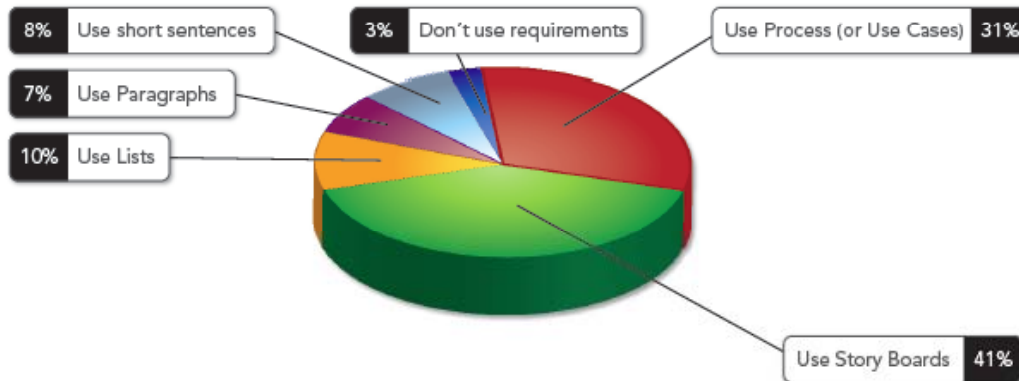
About Acceptance Criteria

- **Acceptance criteria must be tangible and measurable**
 - Allows business analysts to express what the customer wants
 - Developer needs to be able to write an acceptance test based on acceptance criteria.
- **Basic categories of acceptance criteria**
 - Format (e.g. UI must be section 508 compliant)
 - Functionality (e.g. Flight cancellations must be performed manually)
 - Reliability (e.g. 99.999% uptime)
 - Performance (e.g. screen loads within 5 seconds, list loads 500 records in 10 secs, visual feedback on item selection occurs in less than 200 ms)
 - Capacity (e.g. 100 concurrent transactions, 1000 transactions per second)
 - Security (e.g. web audit trail of all transactions)
- **Verify ramifications of acceptance criteria with the customer**
 - Tracking of detailed web interactions means need to store 10TB per month
 - Manual flight cancellation has both positive and negative impacts
 - 99.999% uptime means redundant system architectures and hot swapping
 - Slow visual feedback means examination of frameworks for performance issues

Stories or Acceptance Criteria Can Take Many Forms

- “Requirements must be immediately consumable in Agile. If you are writing English prose, and paragraphs, developers will skip requirements all together and make their own.”
- “Requirements that include visual assets (such as business process diagrams, use cases, user interface mock-ups, and data relationships) require less interpretation from project teams and are more accurately leveraged for project direction...”
- “We found that pictures are the right type of requirements.”

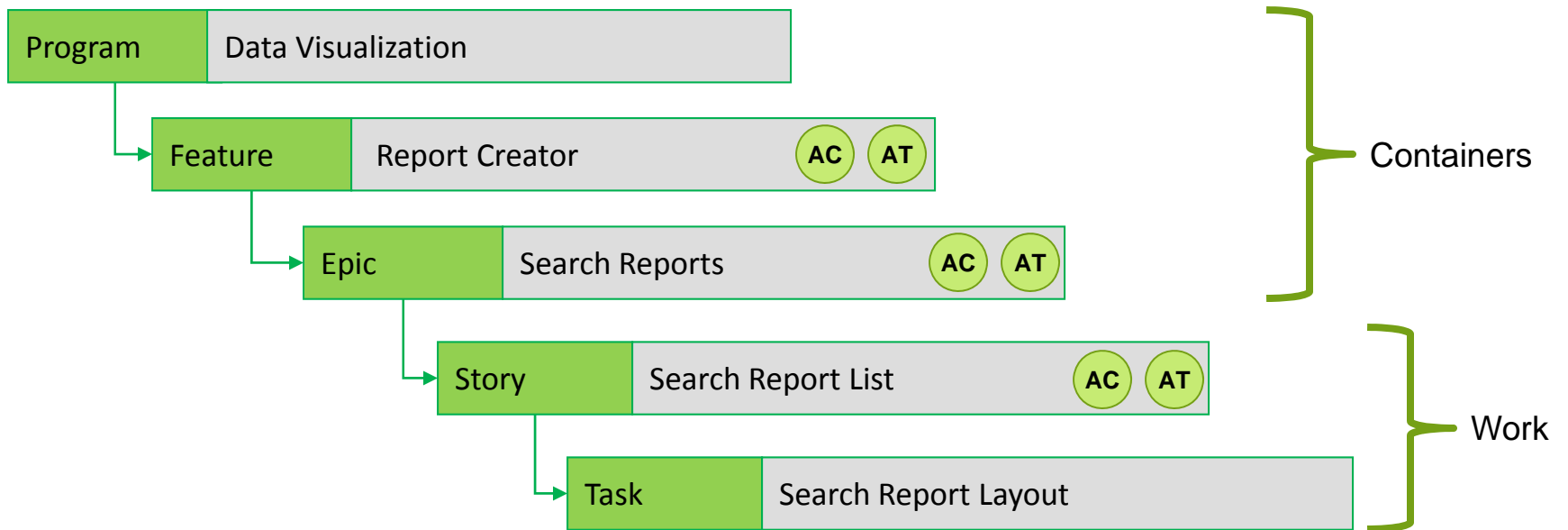
How should requirements be best expressed in Agile projects?



From The State Of Business Analysis In Agile Projects Survey © 2010 requirements.net.

Agile-In-The-Large Artifacts

- Agile-In-The-Large is our process for scaling Agile in large organizations
- In terms of artifacts, you need to have more than stories and tasks
- Programs, features and epics allow you to bundle stories



Design Exploration & Prototyping

Purpose, Activities, Artifacts

Purpose

- Determine the technical feasibility
 - Can and how do we build it?
- Determine the product differentiation
 - Can we build it better than the next guy?
- Explore functional and form variations
 - How do we make it useful and usable?
- Develop a product vision and roadmap
 - Does everyone know what we are building?
 - How do we partition it so that we can build it reasonably quickly?
 - How are we partition it so that we can roll it out and make money?

Parallel Explorations

Form and Function Exploration

- Do form and function analysis
- Sort and prioritize personas/tasks
- Explore primary layouts and navigational models (e.g. taskflow, wireframe, sketches)
- Develop secondary layouts as needed
- Develop paper prototypes

Software Exploration

- Do technology analysis
- Develop architectural concepts
- Identify items that merit proof-of-concept prototyping (e.g. usually risk or unknown items such as structure, api, security, performance, scalability, platform issues)
- Develop software prototypes

Functional Design Questions

- Usefulness

- What is the essential value of the feature/product?
- How can we make this value obvious or visible?
- How do we balance between complexity and visibility?

- Usability

- Do we understand the primary task flows?
- Do we understand the secondary task flows?
- How should we streamline the task flows?

- Priority

- Tablestakes features: Satisfaction versus dissatisfaction
- Differentiating features: Speed versus innovation
- Nice-to-have features: Richness versus complexity

Form Design Questions

- Legacy

- How much innovating can we really do or are we kidding ourselves?
- What are our customer's saying about our legacy form?
- What are the niche competitors doing?

- Differentiation

- MP3 player was the functionality
- iPod was the form play

- Persona form bias

- Salesman suggests “portable”
- Doctor suggests “portable” and “pen”



Software Design Questions

- Do we understand how this would be implemented?
 - Is there more than one way to do this?
 - Do the requirements dictate specific technology choices?
- Do we see any immediate ‘gotchas’?
 - Security, performance, scalability, conformance, data access, data manipulation
 - We tend to defer the unsolvable until it’s too late
- What is our level of expertise in this area?
 - Have we done anything like this before?
 - Has anyone else done it before and can we reuse their solution?
 - Do we even know enough to have an opinion?
- Is this solution worth building?
 - Too complicated for the value we are delivering?
 - How does it fit with the rest of the product?











Design Techniques

- Ways of exploring and expressing design

Design Techniques	Function	Form	Software
Task Sorting (Functionality Sorting)	■		■
QFD – House of Quality (Functionality Sorting)	■		■
Spreadsheets (Computation Rules)	■		■
Decision Tables and Trees (Logic Rules)	■		■
Organizational Diagrams (Block)	■	■	■
Process Diagrams (Flowcharts)	■	■	■
Relationship Diagrams (Object, Entity, Hierarchy)	■	■	■
Spatial Diagrams (Maps)	■		
Temporal Diagrams (Interaction)	■	■	■
Low and High-Fidelity Prototypes	■	■	■

Task Sorting By Persona Value

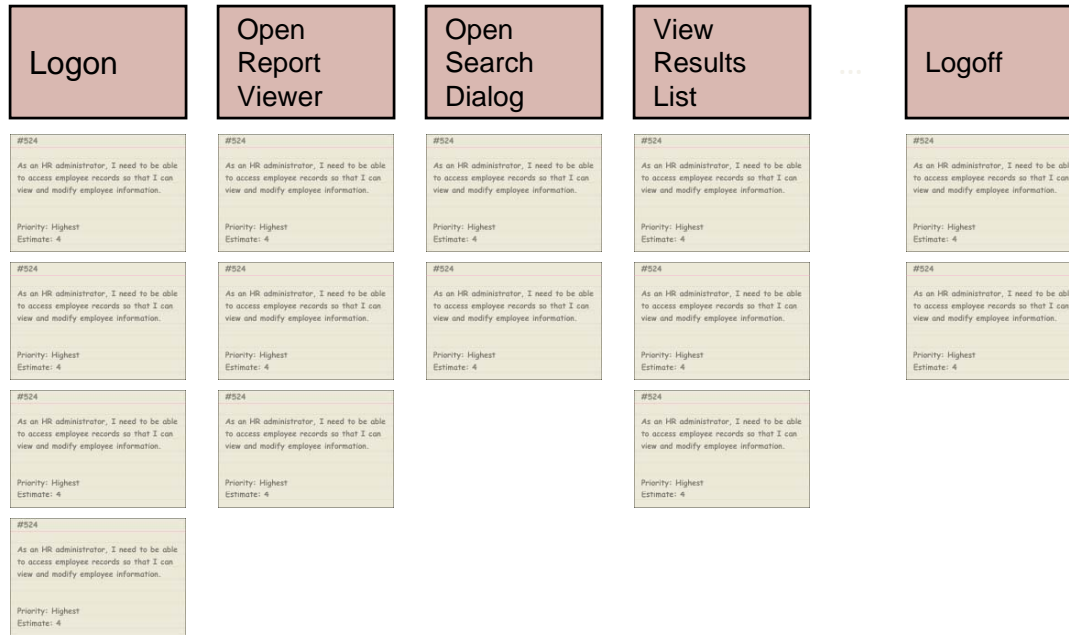
- Task sorting drives architecture and 'look & feel'
- Task sorting prioritizes development
- **Kano** model can be applied to sorting tasks
- Persona-to-task mapping

Category	Description	Persona A	Persona B
Must Be	Taken for granted when filled. Dissatisfaction when not filled.		
One Dimensional	Satisfaction when filled. Dissatisfaction when not filled.		
Attractive	Satisfaction when filled. No dissatisfaction when not filled.		
Indifferent	Neither satisfaction or dissatisfaction		
Reverse	Satisfaction for some personas Dissatisfaction for other personas		

Task Sorting By Time

Time

Search
Reports
Task Flow



Taskflow

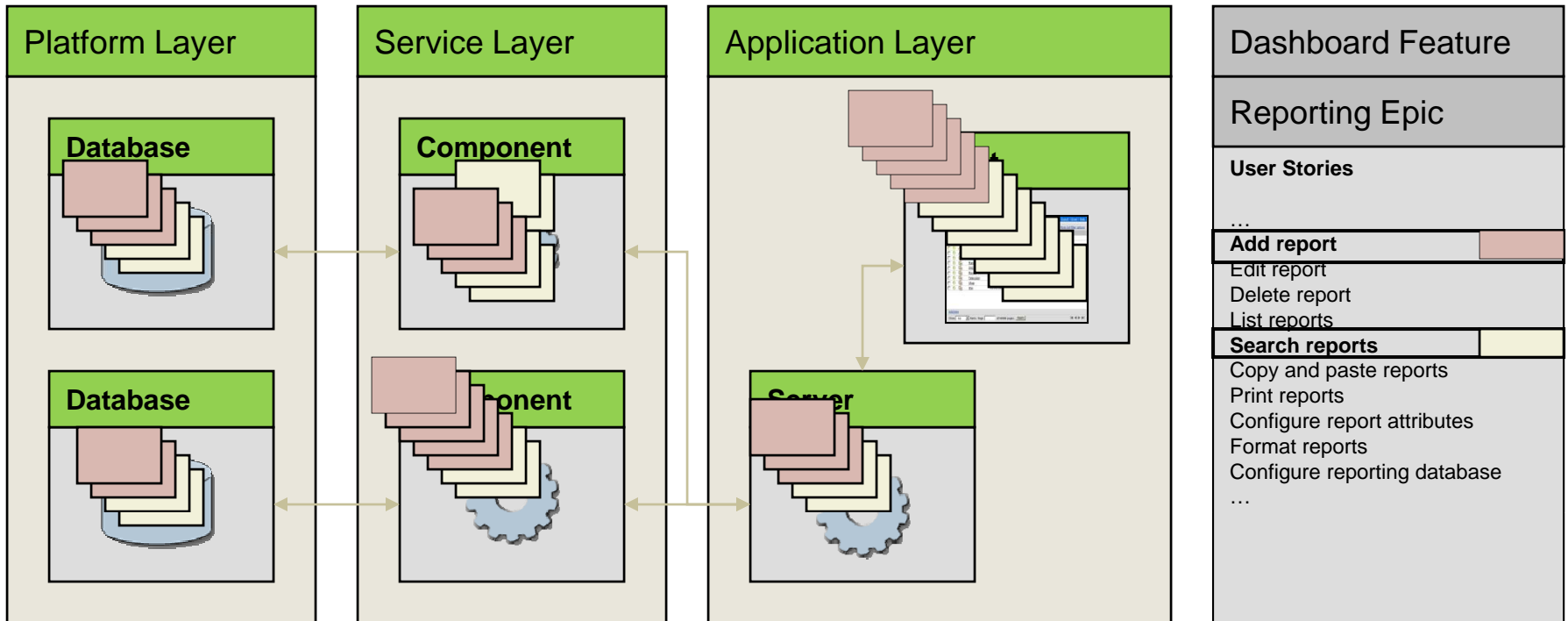


Taskflow



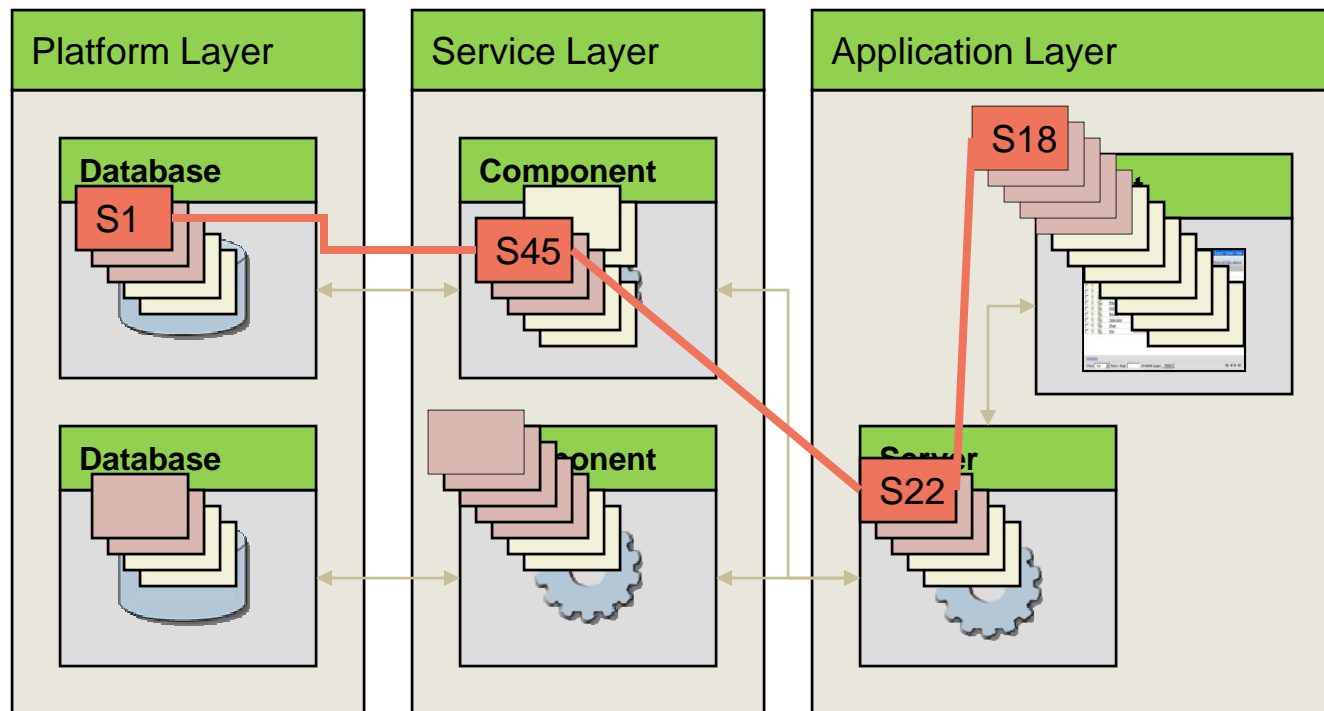
Task Sorting Drives Architecture

- Understand what you need in your architecture to make customer happy
- Identify bottlenecks – N stories use this one thing too often
- Identify efficiencies – N stories use the same functionality
- Identify work load based on architecture



Task Sorting Lets You Show Progress Through Architecture

- Create a working 'thin slice' through the architecture
- Also called tracer bullets, essential use cases
- Allows you to test functionality and show progress to stakeholders
- Example: 'S1, S45, S22 and S18, we can show a simple search...'



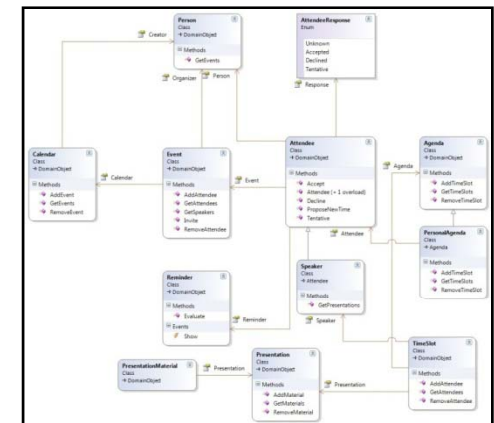
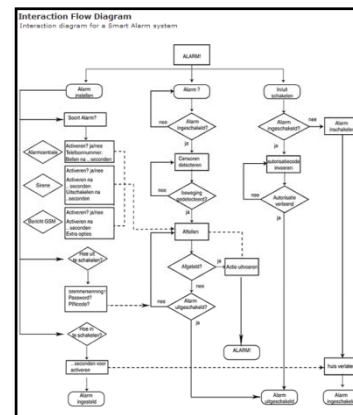
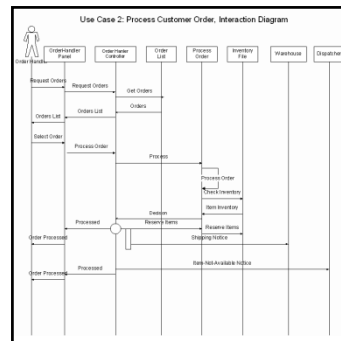
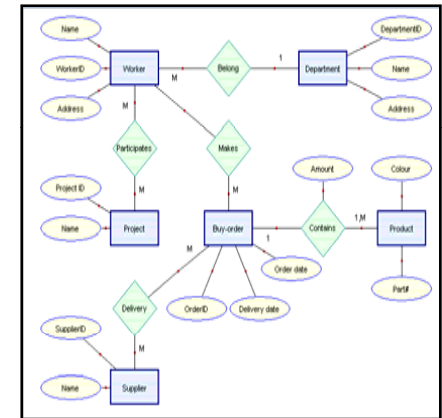
Architecture Visualization Techniques

- Types

- Process visualizations (flowchart diagrams)
- Temporal visualizations (interaction diagrams)
- Relationship visualizations (object or entity diagrams)

- Value

- High-level and detailed blueprint of the system
- Avoid major unanticipated design flaws
- Identify scalability issues
- Partition workload



Business Logic Decision Table Technique

- Types

- Plain old table!
- Can use Excel

- Value

- Simple way of getting your head around complexity by structuring logic or complex problems
- Captures conditions
- Captures actions
- Captures outcomes
- Captures relationship
- Easy to generate code
- Easy to update

No charges are reimbursed to the patient until the deductible has been met. After the deductible has been met, reimburse 50% for Doctor's Office visits or 80% for Hospital visits. There will be 4 rules. The first condition (Is the deductible met?) has two possible outcomes, yes or no. The second condition (type of visit) has two possible outcomes, Doctor's office visit (D) or Hospital visit (H). Two times two is four.

Conditions	1	2	3	4
1. Deductible met?	Y	Y	N	N
2. Type of visit	D	H	D	H
Actions				
1. Reimburse 50%	X			
2. Reimburse 80%		X		
3. No reimbursement			X	X

Source: http://web.sxu.edu/rogers/sys/decision_tables.html

Primary UI Exploration Technique

- We tend to design the first thing that comes to mind
- Orderly approach to visual exploration
- Helps to explore the user's organizational or mental model

Organizational model	View houses for sale
Location or map views Used to show visual relationships between various display objects. (e.g. physical maps, network topologies, drawing, process, desktop)	On a map
Alphanumeric views Used when tabular comparisons, search, filter are important (e.g. spreadsheets, alarm managers, email lists)	In a list
Time views Used when time is an important relationship (e.g. project management, calendars, planners, animation)	On a map with a time slider
Category views Used when the category is the important relationship (e.g. models, departments, organizations, classifications)	In a list by price
Hierarchy views Used when seeing and understanding parent-child relationship is important (e.g. tree structure-based applications like Explorer or Outlook)	In a tree by city
Based on Richard Saul Wurman's LATCH model. Read his book "Information Anxiety" for more information.	

Primary Layout Example

- List mental model
- Based on category
- Search criteria then to list

- Map mental model
- Based on location
- Search criteria coupled to map

The screenshot shows a real estate search interface. On the left is a search filter panel with various options like 'Price Range', 'Type', 'Storages', 'Bedrooms', 'Bathrooms', and 'View'. Below the filter panel is a list of search results. Each result includes a price, address, and a small image. A large green arrow points from the search filter panel to the list of results.

The screenshot shows a real estate search interface with a map view. The map is surrounded by search filters on the left and a list of search results on the right. A green arrow points from the search filter panel to the map.

Prototyping

- Types of prototypes

- Fidelity should be as high as it needs to be; pay now or pay double later
- Low-fidelity paper prototypes (e.g. Paper or PowerPoint)
- High-fidelity animated prototypes (e.g. Flash)
- Software prototypes

- Use low-fidelity or high-fidelity prototyping when...

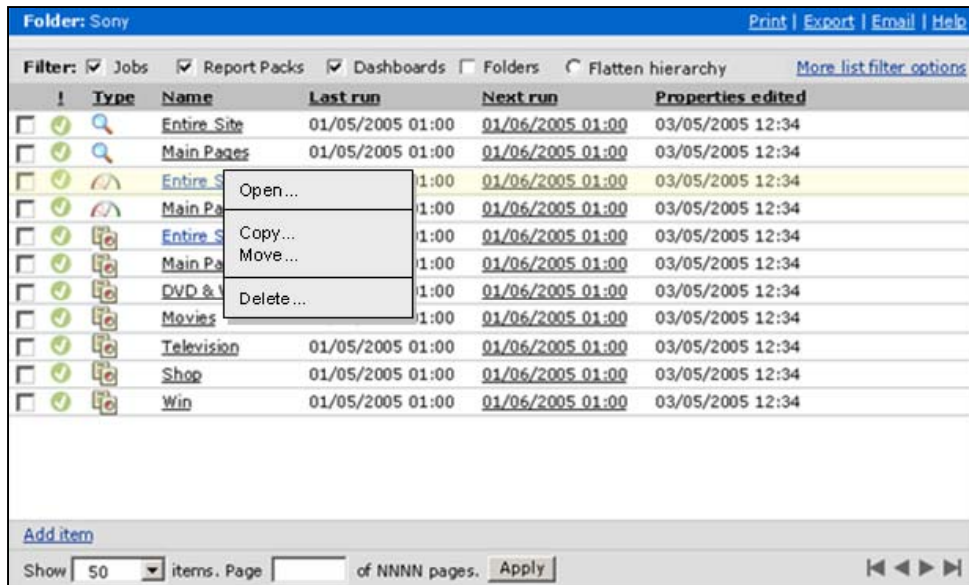
- Business (e.g. new functions, integrations)
- Functionality or Capability (e.g. new functions, features, workflow)
- User Experience (e.g. organization, navigation, appearance, accessibility, usability)

- Use software prototyping when...

- Structure (e.g. technologies, integration, 3rd party integration, scalability)
- Performance (e.g. transaction rates, data storage access and retrieval, response times, throughput, concurrency)

Low Fidelity Prototyping

- UI picture is worth a thousand stories
- UI prototyping helps you design system
- Makes functionality concrete and tangible
- Used to validate concepts with customers
- Cheaper than code



Stories In This Picture

1. Open business area
2. Print business area
3. Email business area
4. Export business area
5. Filter business area
6. Page list
7. Open help
8. Add report
9. Add dashboard
10. Add report configuration
11. Add dashboard configuration
12. List reports
13. List dashboards
14. List report configurations
15. List dashboard configurations
16. Open or view report
17. Copy report
18. Move report
19. Delete report
20. Open or view dashboard
21. Copy dashboard
22. Move dashboard
23. Delete dashboard
24. Open or view report config
25. Copy report config
26. Move report config
27. Delete report config
28. Open or view dash config
29. Copy dash config
30. Move dash config
31. Delete dash config

Low-Fidelity Prototype Example

- Low-fidelity prototype

- Initially rough and then later refined drawings
- Interactive branching allowed walkthrough
- User model, task model, task flows
- 3 structure and navigation alternatives
- 2 visual form alternatives

- Concept iterations

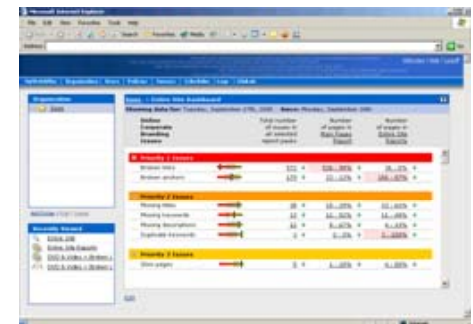
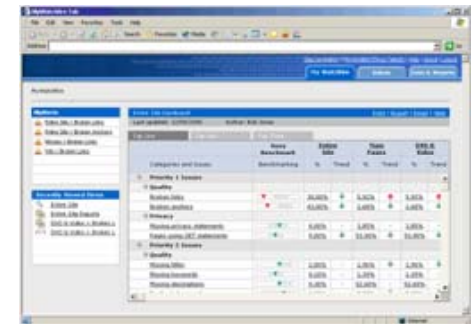
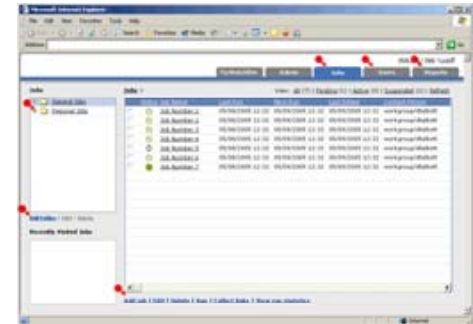
- 6 iterations (expanding from 8 to 48 screens)
- 3 sprints
- 3 internal / 2 external customer sessions

- Detail iterations

- 3 iterations (148 screens)
- 8 sprints
- 3 internal / 1 external customer sessions

- Investment

- Less than 2% of overall effort



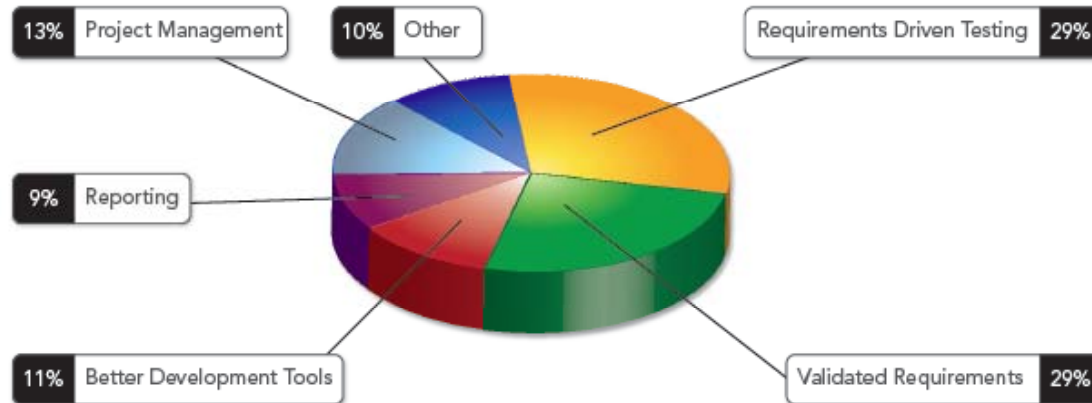
Design Verification

Purpose, Activities, Artifacts

Purpose

- To validate usefulness or usability of concepts with “real” users
- 80% of the usability problems were found with 4-5 participants...
 - Ehrlich and Rohn, “Cost-Justifying Usability”, 1994
- 29% of Business Analysts consider it a best practice...

What are the processes that best improve business and IT alignment in Agile processes?

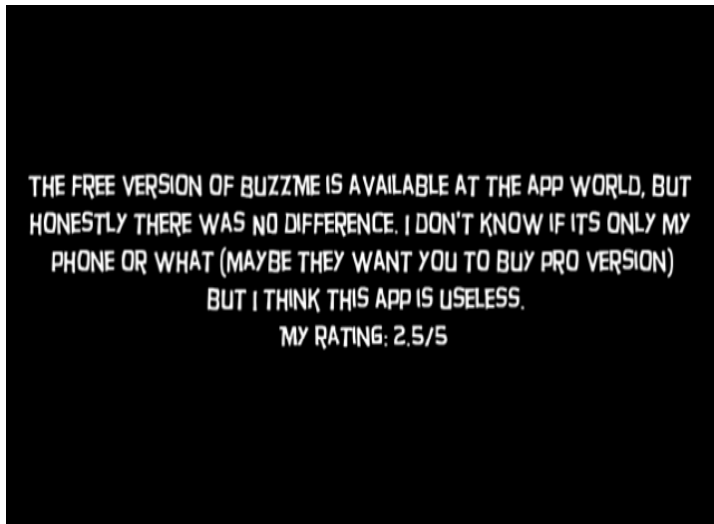


From The State Of Business Analysis In Agile Projects Survey © 2010 requirements.net.

Effective way of managing/prioritizing feature creep...

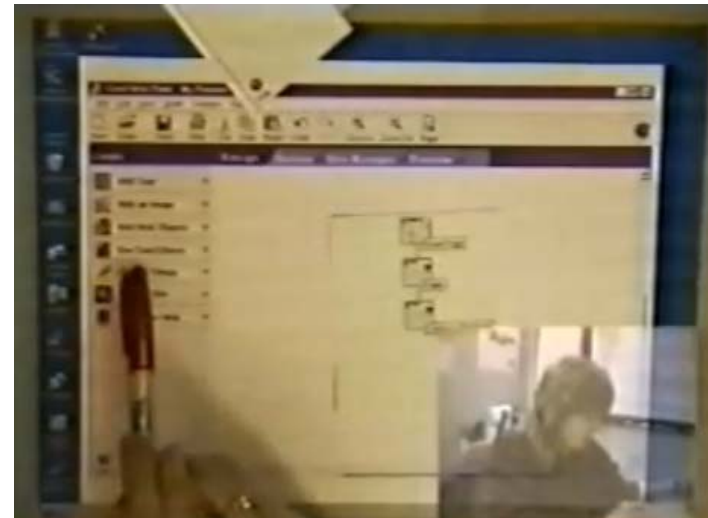
Types of things to test

- **Usefulness**
- Most important!
- Avoid building features nobody wants
- Wasting money and time



<http://www.youtube.com/watch?v=1G6xHeFvoWM>

- **Usability**
- Fix product issues
- Create satisfied customers
- Discover new features



“I would do ‘change things’ ...”

“I would go to ‘page layout’...”

“I would click “change site” and see what happens...”

<http://www.youtube.com/watch?v=ppnRQD06gg>

Typical Approach

- Planning
 - What do I want to test?
- Protocol development
 - how are we going to test: order, method, stimulus materials
- Conducting the test
 - Web-based or live facilitation of test
- Reporting
 - Collation, sorting, prioritization of results

Conclusion

- Envisioning is a light-weight, iterative activity that couples user-centric design with software exploration
- Allows you to:
 - Explore product alternatives (value, usefulness, usability)
 - Evaluate technology alternatives (feasibility)
 - Make requirements tangible
 - Establish acceptance criteria
 - Validate solutions with customers
 - Prioritize development based on customer priorities
 - Establish a clear product vision
 - Establish a development roadmap

Thank You!