Agile and Usability

MB

Agile Manifesto

- "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 t
  - Working software over comprehensive documenta
  - 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**."—The Manifesto for Agile Software Development
- Beck, Kent, et al. *Manifesto for Agile Software Development*, 2001. Retrieved October 4, 2009.

e.g., In forums, emails, wiki and Trak

Some characteristics...

- Agile teams are typically cross-functional—with more emphasis on the skills of the team members than on organizational roles—and small—comprising less than 10 people.
- Development takes place through a series of iterations—typically lasting between 1 and 4 weeks.
- Stakeholders **prioritize requirements** according to their "business" value, and working software is the primary measure of how a project is progressing.
- Each agile team includes a "customer" representative, who is available to answer questions regarding the problem domain.

Some characteristics and challenges to UX

- UI design: a consistent user experience across features, without changing UX redesigns' regularly requiring users to relearn how to interact with the software.
- Establishing **broad design principles** early in the development process, understanding user characteristics and their probable demands on the software, context, **tool ecology** and designing in a way that anticipates these—without committing to them—can maximize the degrees of freedom for a design

Some characteristics and challenges to UX

- Opportunities to reuse existing design components, or patterns.
- UX designers to work **in parallel** with the rest of an agile team, developing the UX design slightly in advance of the next development cycle.
- In later design cycles, being able to refactor the user experience can help keep an application's design clean and flexible.
- For software with multiple target user groups, care must be taken to avoid representing one user group over others, particularly if the customer representative is not from the primary user group – e.g., using personas and scenarios to maintain **Consistency**!

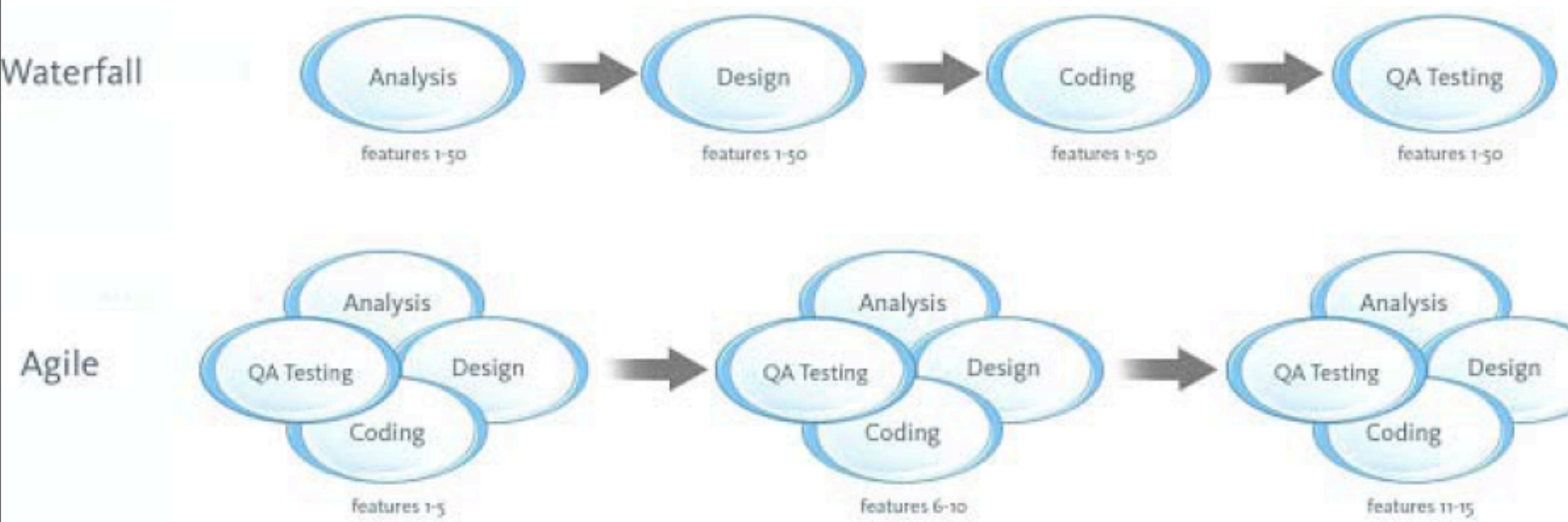
Some characteristics and challenges to UX

- UX designers must be individuals, who **are able to make decisions** about the user experience and plan within a broad framework (cf. the semantic turn idea)
- Distributed teams: teleconferences and video conferences can come into question in keeping the team coordinated but this is no key solution, there are studies that show that it is still very hard to work in Agile way in distributed teams!
- Use of **Rapid prototyping for user experience** — e.g., paper prototypes— can let you elicit rapid feedback on your designs.
- Whenever possible, **test the usability of your designs** with representative users.
- Heuristic evaluation techniques can provide high value for a minimal outlay of time and resources, and they can help other development team members understand the software's user experience.

Some characteristics and challenges to UX

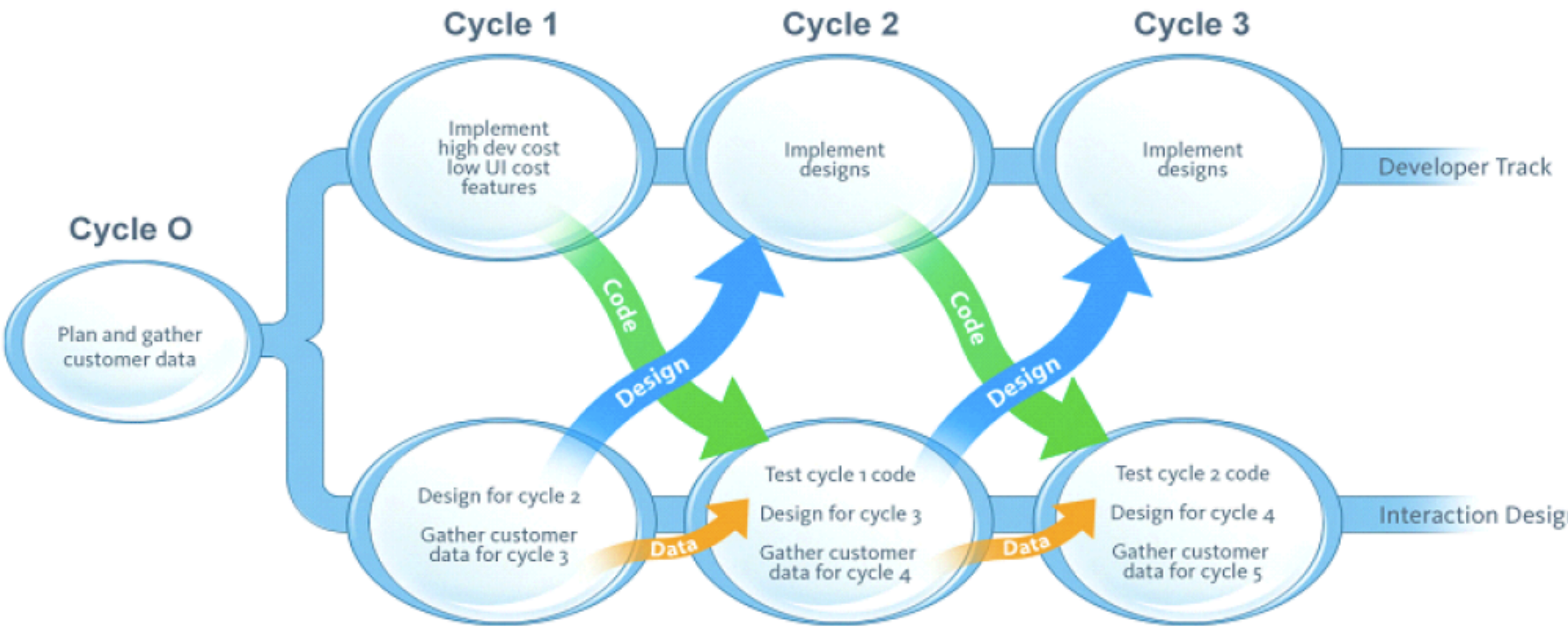
- **Ensure that your testing focuses tightly on the areas of the user experience that a development release affects.**
- Target user group as test participants.
- Agile methods challenge UX professionals to be more flexible and adaptable, to work more closely with developers, and to have closer contact with a product's users.
- They force UX designers to work more closely with other participants in the development process than traditional development methodologies do.

Process....
from Adapting Usability Investigations for Agile User-centered Design by Desirée Sy



Metropolia

Process....
from Adapting Usability Investigations for Agile User-centered Design by Desirée Sy



Metropolia

Cycles…

- Usability investigation activities in Cycle Zero
- Cycle Zero is the brief requirements-gathering phase at the start of the project.
- Usability investigation activities depend on whether the product is the next release of an existing product or completely new.
  - They can include the following activities:
  - Gathering data to refine or bone product- and release-level goals. Facilitating the alignment of all team members' understanding of these goals, so they constitute a shared vision.
  - (For a completely new product) Interviewing or conducting contextual inquiry during customer site visits for market validation. Preparing high-level exploratory designs for market validation.
  - Based on these data, deriving the **design principles that inform and guide design decisions for the product**.
  - (For an ongoing release) Analyzing and summarizing prior contextual inquiry and **usability test data.**
  - Based on these data, elucidating **release-level design goals** to inform and **guide design decisions** through all iterations.
  - (For a completely new market or capability) Developing brief and vivid descriptions of target users and workflows (light **personas and scenarios**) from investigations.

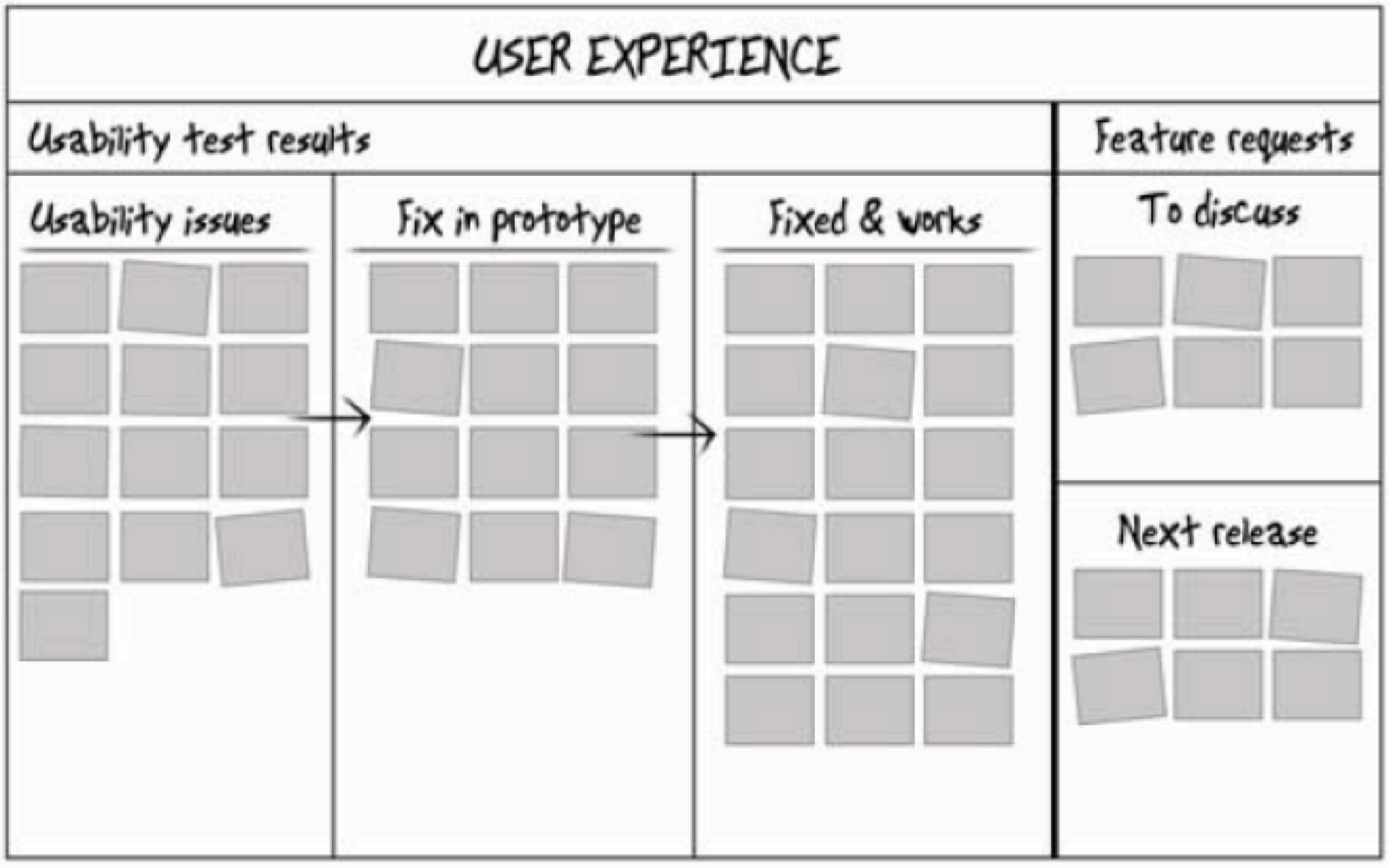
Design granularity changes...

- **Design chunking: Breaking designs apart into cycle-sized pieces**
- To deconstruct a large design into smaller pieces, it is essential to start with well-defined design goals and to understand the **high-level design intent**. Our design goals are derived from observation, which is why contextual inquiry plays a critical role in our Agile UCD process. Each design chunk lets us progressively achieve a subset of the **design goals**
- The **priority and sequence** of the design chunks is determined by:
  - what we can validate at any given time in the product lifecycle.
  - Examine the full list of design goals, and decide which we can attain with the current resources within a cycle's length.
- There is also an **ordering dependency**.
  - Components build on one another, so early design chunks must be low-level and fundamental —design attributes that will not change as more design chunks are added on top of them.

Design granularity changes...

- The types of design chunks that usually are completed in later cycles include the following:
  - Prototypes that require an implementation or technology that hasn't been completed yet.
  - Design chunks that provide workflow-level, rather than operation-level, functionality.
  - Design chunks to support any investigation of a discoverability or learning goal, such as the design of how a user will access a new function.
- Since these are designs that depend on the first experience of a user. Need to replicate them. These prototypes should be incorporated into a copy of a working version to avoid task bias.
  - Design chunks that are hubs for other designs
- They are simply design components that can be prototyped, iterated, and validated within Agile timeframes.
- In design chunking,the high-level design considerations are not ignored; instead, it is worked toward them in cyclesized steps.

Process....
from Adapting Usability Investigations for Agile User-centered Design by Desirée Sy



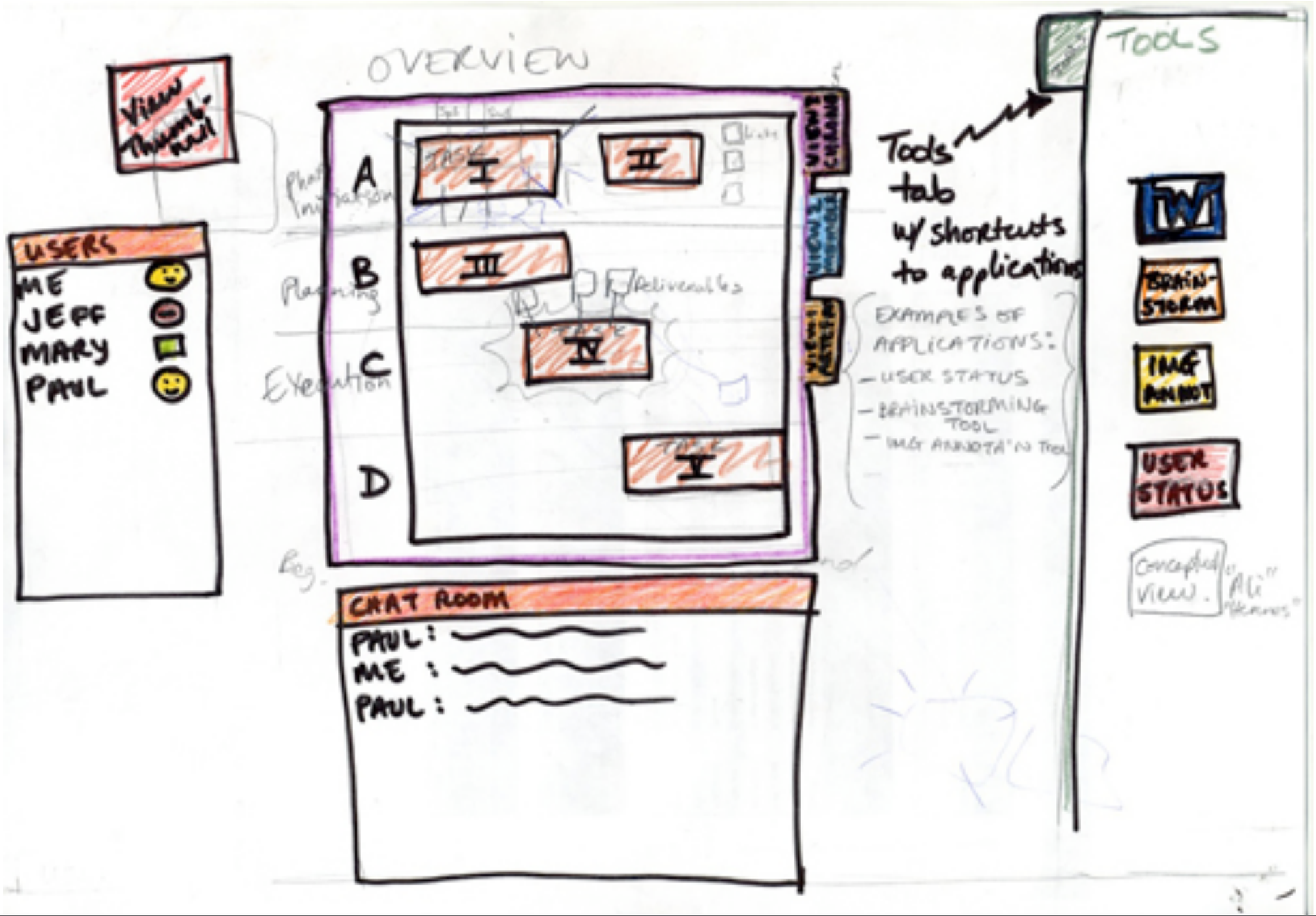
Metropolia

Design history WHY?

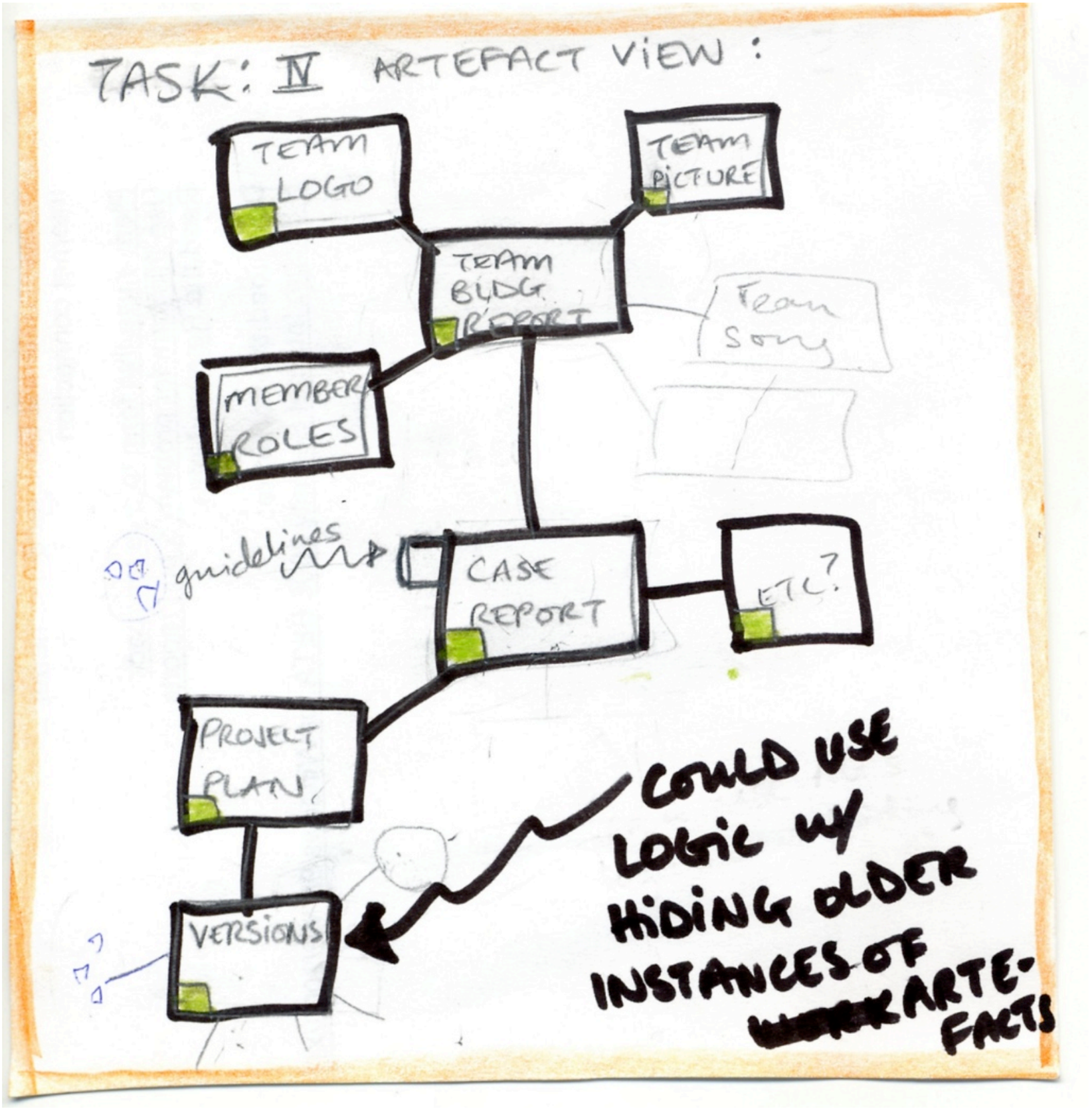
- The purpose of the record is principally to avoid "design thrash," where design decisions are accidentally re-visited, particularly between versions, or if a new User Experience Team member is added to a project.
- One document for each implementation chunk.
- The document is written in a medium that is easy to update (such as a Wiki page).
- Information in a design history document can include the following:
  - Design goals and a brief description of the problems that the design addresses.
  - A high-level description of the design, including rough sketches, and a pointer to the last design prototype.
  - Links to related design history documents.
  - A reverse chronology of the design iterations, including the reasons for the design changes, and design limitations and constraints defined by usability investigation data as design work proceeds.
  - Relevant usability investigation data are recorded in this chronology. This section of the document is written as design cycles progress. The oldest entry in this design chronology describes technology constraints.
  - The names of the associated feature cards.
  - As working versions are completed, the design chronology is extended to include additional workflow information and links to bugs related to changes in design, or unexpected uses.

Example...blaa but it is KP-Lab project stuff

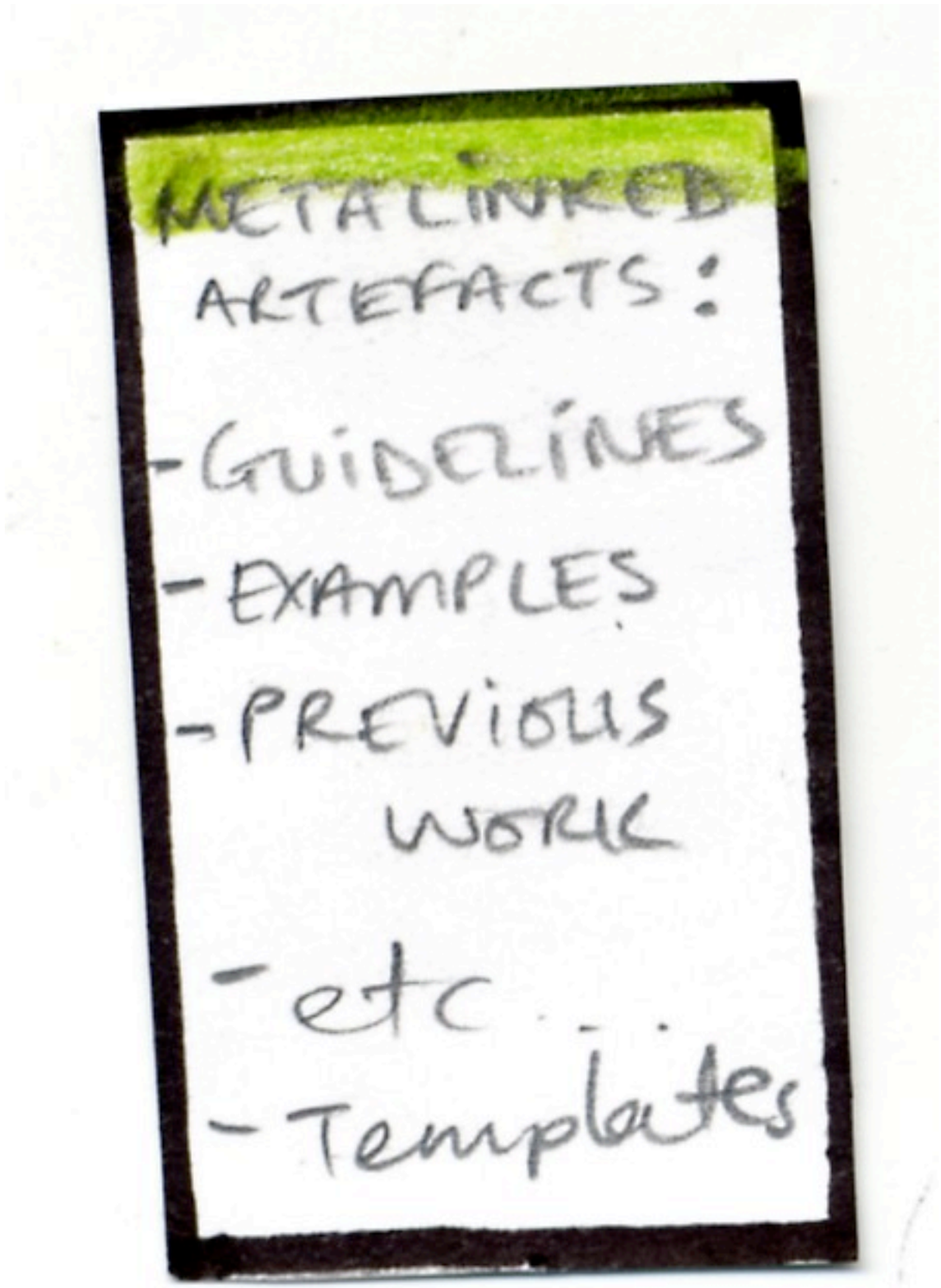
- Starting mock-ups, following the high level principles



Metropolia


TASK: IV ARTEFACT VIEW :
TEAM LOGO
TEAM PICTURE
TEAM BLDG. REPORT
TEAM SONG
MEMBER ROLES
guidelines
CASE REPORT
ETC.?
PROJECT PLAN
VERSIONS
COULD USE LOGIC W/ HIDING OLDER INSTANCES OF WORK ARTE-FACTS

Metropolia


METALINKED
ARTEFACTS:
- GUIDELINES
- EXAMPLES
- PREVIOUS
WORK
- etc...
- Templates

Design processing info of one chunk

- http://kplab.evtek.fi:8080/wiki/Wiki.jsp?page=M24ProcessPlanningRequirements

Usability tests of chunks and in context

- http://kplab.evtek.fi:8080/wiki/Wiki.jsp?page=Usability#section-Usability-CategoryOfFeedbackTestingData

Design document

Working Knot:
WK1 Shared Space and Common Tools
Sub knot: Semantic and free term search

High Level Requirements:

HLR1.4 (DO 1)	Users can search artefacts within and outside the shared environment using full text, metadata or domain ontologies.
HLR8.6 (DO 8)	Users can search the content and metadata using full text or semantic metadata search for planning and reflecting on activities.

Description:

After logging in to the KP Environment and navigating into any view of it, user wants to search for knowledge objects available in the KP Environment based on some free terms (text). User opens the search tool from the menu bar and types in search terms into the search field. She can constrain the search in two ways:

1. Whether to search all or a subset of shared spaces
2. In which parts of the knowledge objects to search (full text content, title, description or all metadata).

After user has entered a search terms or selected a constraint, system performs the search by using variations and extensions of string matching to find the relevant knowledge objects in the KP-Lab content and knowledge repositories. The search results are displayed as a list. User can then open any of the content items shown in the results list by double-clicking on the link associated with the title of the item.

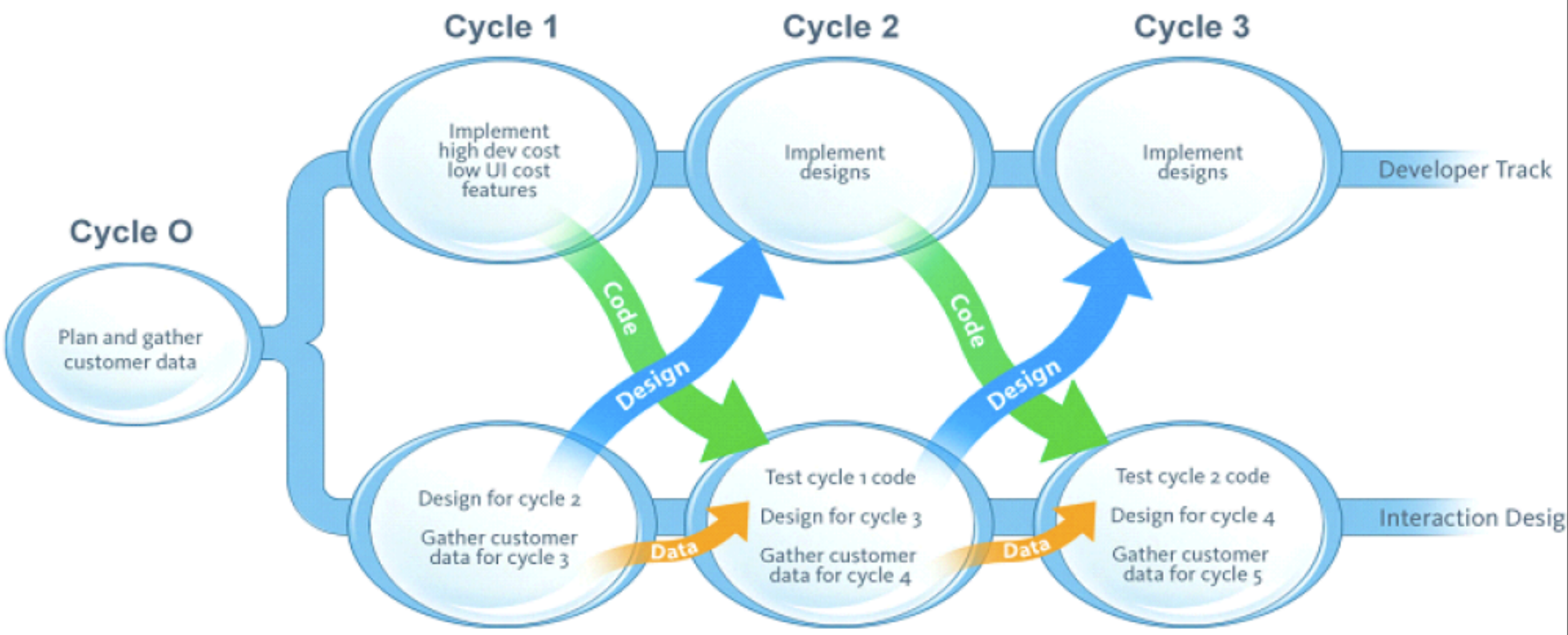
Figure 1 presents a mock-up of the search GUI. The three parts relevant for the basic free term search are on the right hand side. The upper part provides the possibility to select from where to search (i.e. which shared spaces) and from where in the content items to search (i.e., full text content, description, title, and/or all metadata). The middle part displays the summary of search criteria specified. The lower part lists the results of the executed search query.

Metropolia

Some interesting sites and blogs:

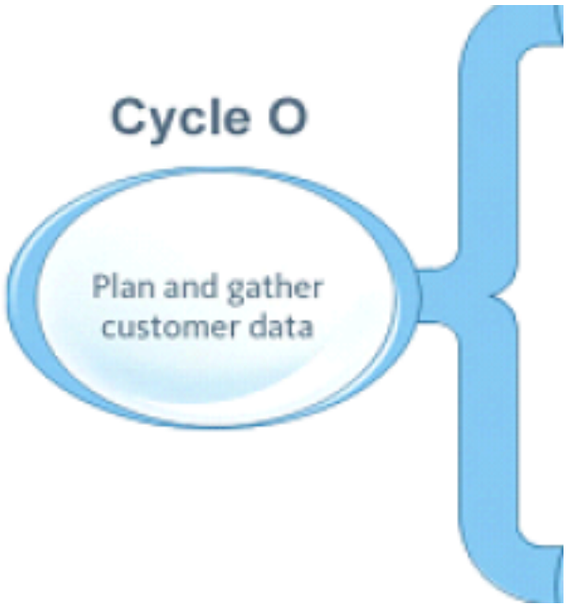
- http://agileproductdesign.com/blog/emerging_best_agile_ux_practice.html
- http://agileproductdesign.com/presentations/usage_to_user_interface/index.html
- http://www.uxmatters.com/mt/archives/2009/10/can-ux-be-agile.php
- See description with examples about UX & Agile methods emerging to fuse them ...http://www.upassoc.org/upa_publications/jus/2007may/agile-ucd.html, Adapting Usability Investigations for Agile User-Centered Design
- http://agileproductdesign.com/blog/emerging_best_agile_ux_practice.html AgileProductDesign.com

Usability in Agile - broken apart



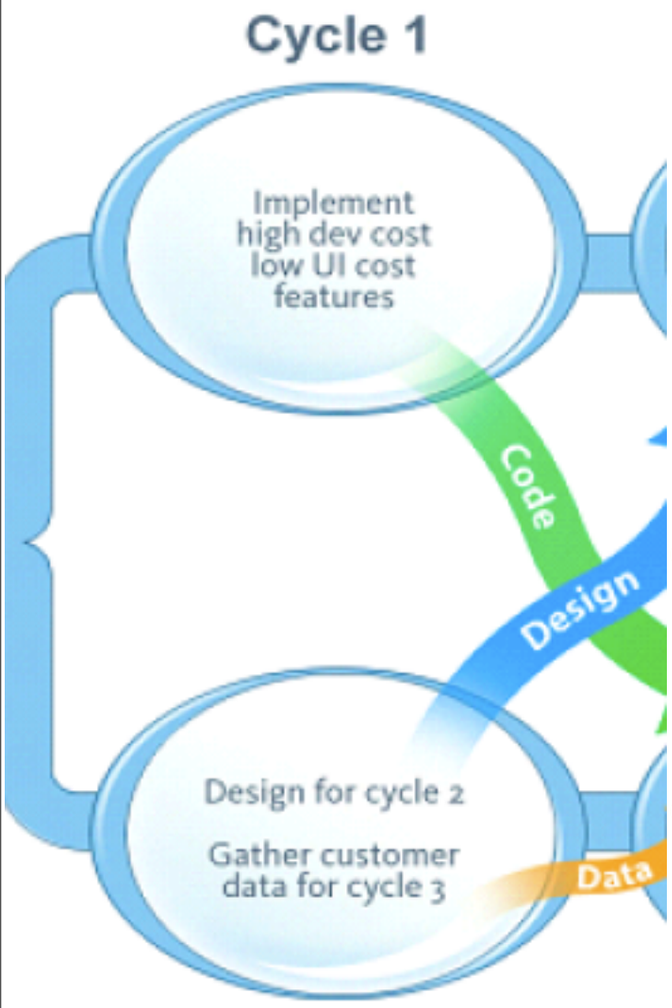
3 parallel tasks going on:
- To iterate on designs, usability tests on mock-ups one cycle ahead of developers– after OK, implement
- Continue contextual inquiry for the future components
- Usability test the implemented working version

Usability in Agile - broken apart



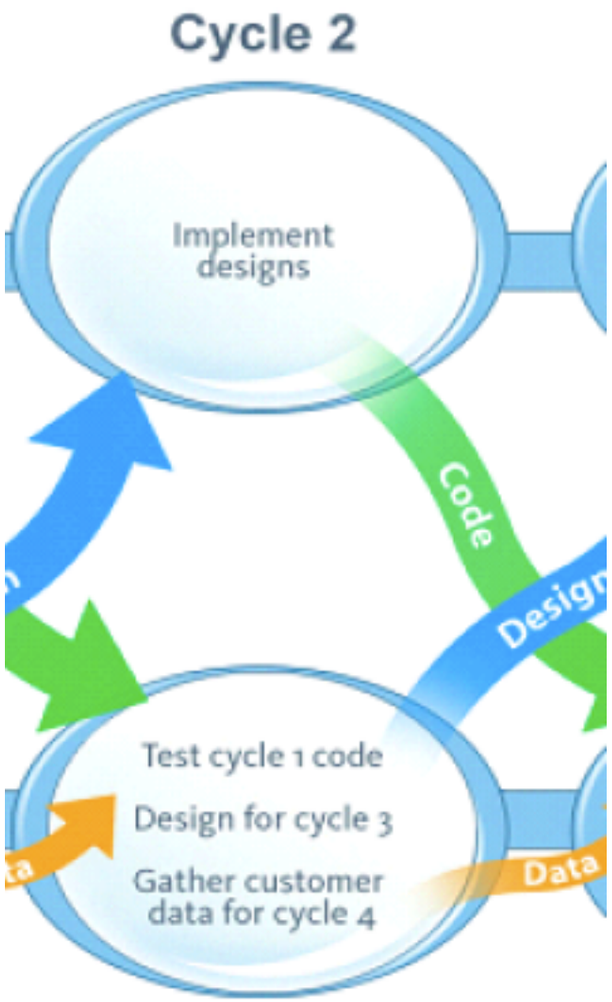
- Gathering data to refine product- and release-level goals – requirements and the needed **usability criteria** that should be acquired
- Result: set of design principles and objectives as well **usability criteria to focus on** (in more abstract level)
- First component features and functions
- Its interaction flow

Usability in Agile - broken apart



- Prototypes (i.e., mock-ups): "usability testing" (e.g., the wiki commenting example above)
- The mock-ups may need more cycles themselves than just one
- Use colleagues to test the interaction logic
- Use developers to comment on the interaction logic (they usually spot inconsistencies or omitted important steps, attributes etc.

Usability in Agile - broken apart



- **Prototyping and usability testing** for Cycle 3 designs using the requirements information gathered in Cycle 1.
- Requirement elicitation for Cycle 4.
- **Usability testing the implemented working version from Cycle 1** in field
- And so it continues

What changes in the usability part

- As the design is in small chunks (determined by the dependencies in technical development(!), prioritising according to the users (stake holders' needs) and taking into account the complexity for the UX design - the chunks are some kind of best compromise
- Usually these are the operational level parts that are know to be the basic interaction components
- Later full workflow design and design that needs to be tested with users for longer period of times, e.g., new ways to use the tools

What changes in the usability part

- The chunks/components have to be such that they can be prototyped and iterated in fast phase
- With the fast cycles there is less time to test full workflow before implementation
- Field use should go along parallel to the other activities - also problematic, its not easy the **schedule real field use according to the design**
- Along the process users get closer in their characteristics to the actual users; tasks also proceed in the process closer and closer to the real task and real context

Test in different phases:

- **In early stages** it is colleagues that have some characteristics of the real users who test, but the testing is more on **the operational level** (example faceted search and tagging)
  - These test can be executed in isolation of the actual work flow and context
- External users are invited to test only **mid- to late-stage** design chunks to focus on validating design goals that can **only be determined by an actual user**
- **Such as?**

Test in different phases:

- **Mid stages of the process:**
- In house session about an hour
- Interviews by telephone to get to know the workflow
- Ask the users to bring artefacts (e.g. relevant files related to that workflow)
- The session is begun with the contextual investigation, watching the user walk the observers through the files and demonstrate key interactions
- Usability test of prototypes exploring different stages for **two** design chunks that under designing in the current cycle.
- During the session, it is noted if the made test activities are representative.
- If needed, the tasks will be adjusted for the next tester for the next time

Test in different phases:

- **Late stages of the process:**
- Two hour session at the users context (environment, i.e., in the field)
- Before coming the user is asked to set up some "files" for demonstrating what s/he has been doing with the Beta version since last visit
- All unexpected uses are especially interesting! and should be show to the whole developer team
- An interview is conducted about the future improvements need for the next release
- And user is asked to show some relevant artefacts related to that (e.g. files or whatever is related to that use)
- The use is observed when s/he present the to be designed workflow

What changes in the usability part

- When work flows are available these prototypes are brought to the users into their use context to be tested
- They are also left there for interviewing experiences and new usages the next time users are visited with improved prototype
- Actually these prototypes should be called Beta since they can be left for use
- Sy calls them “design patterns” (a bit misleading but hmm....)
- Emphasis is in the longitudinal experience!
- Interviews are included with observation as well as when coming back (sort of props interview)

What changes in the usability part

- Getting to know users vocabulary
- Using props in the interview (artefacts that users bring along if testing /Interview situation is not in their place or user demoing their use with the artefact) – helps to focus on relevant parts and helps users to remember better what they do, what they feel etc.
- The test with the different components are also analysed as one broader test to have a more holistic idea of the user experience
- **Key idea:** mix different test methods and use face to face situations with users for requirement elicitation and testing

Reporting changes:

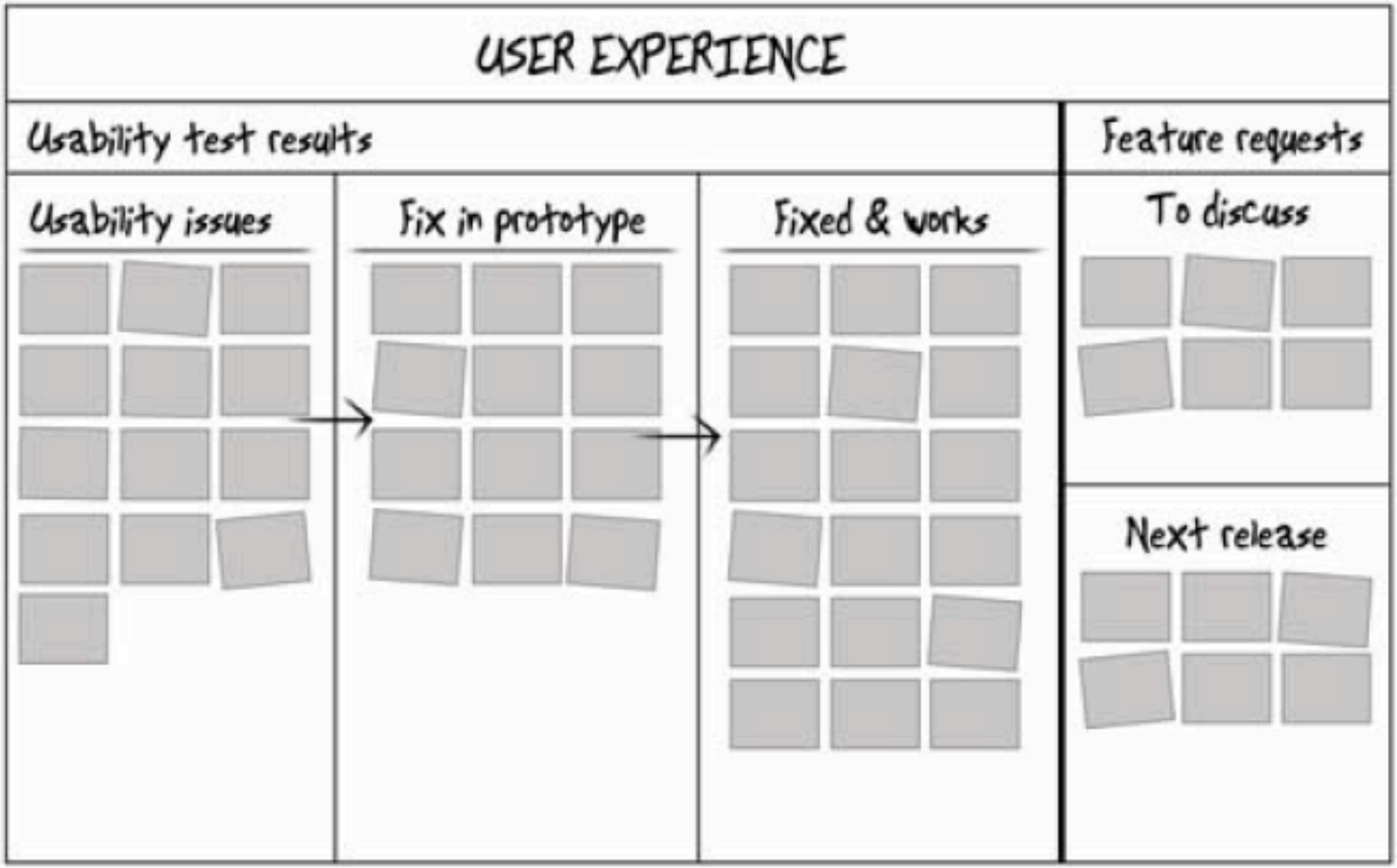
- What is needed to be communicated to the team:
  - Which designs is worked on, and approximately when developers should expect to receive the mock-ups
  - Usability test results and high-level progress for late-stage design chunks
  - Recommendations and fixes for working versions
  - User and task information from external users, especially from field visits
  - The user interface design to be implemented.

Reporting changes:

- Written text is boring and not often read...so
  - Use video montages of usability tests (Next lectures)
  - Use feature cards
  - Technical constraints need to be known (importance of the developers and smooth cooperation between UX designers and developers)

Demonstrations to team

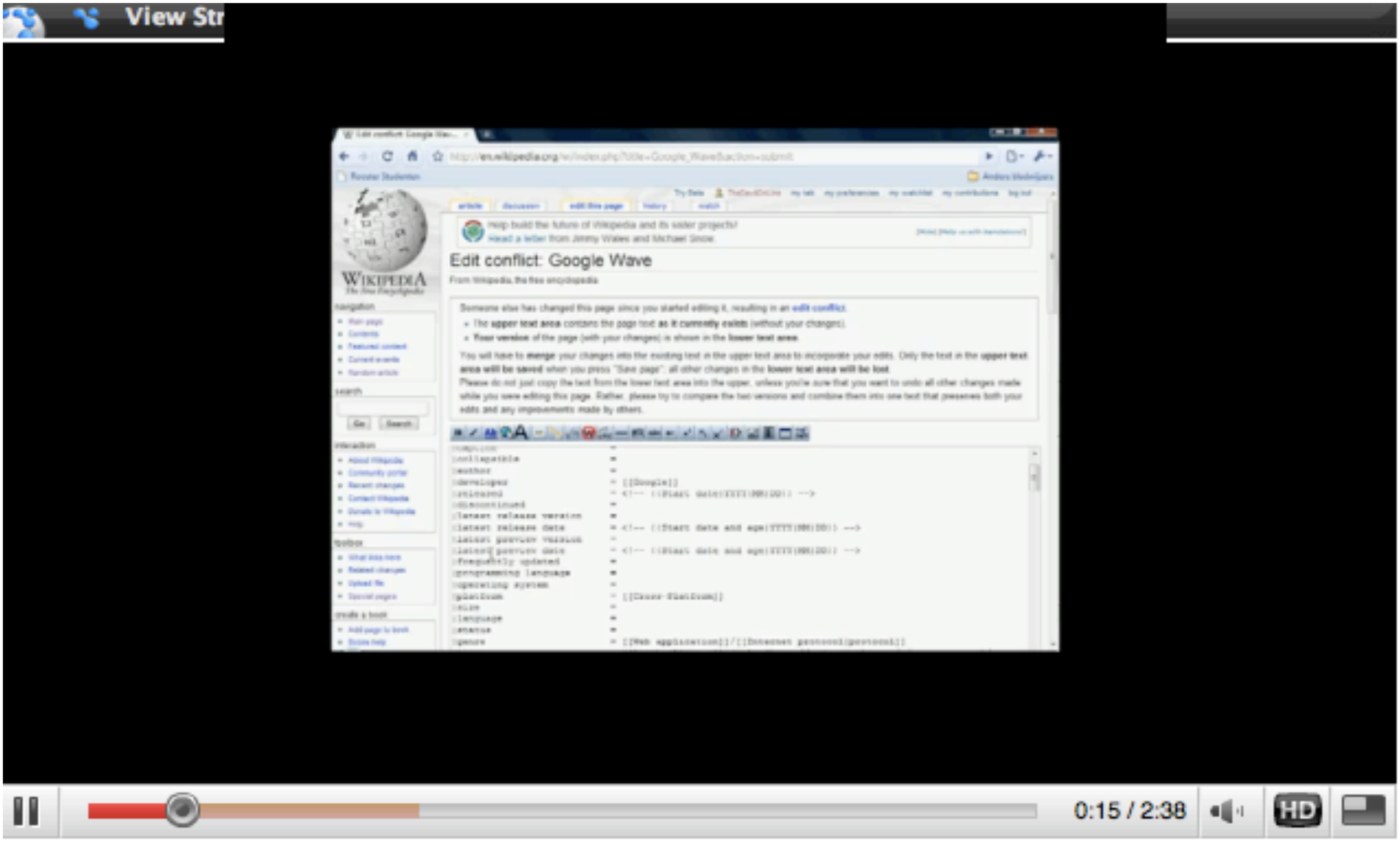
- If possible use demonstrations, video in face to face with the team (SCRUM) about
  - examples of users' work
  - contextual information of users use situation
  - frequent requests
  - unexpected uses
  - major usability problems
  - bugs in beta versions


USER EXPERIENCE
Usability test results
Feature requests
Usability issues
Fix in prototype
Fixed & works
To discuss
Next release

Metropolia

The design/usability/issue cards

- In the cards the same information as above is displayed
- Team decides what to do with the feature requests and unexpected usages
- Bugs into the tracking system
- The issue cards have the fixes, design issues and bugs that have been found and their potential iteration
- Also the usability criteria for the specific issues in the card are presented in the card!



http://www.youtube.com/watch?v=Hk9Xe3f6PAM&feature=player_embedded
http://mickenordin.se/wiki/Wave