Usability in Software Engineering Literature [1]

- Quality Attributes in the index of “Software Architecture in Practice” [Bass, Clements, Kazman, 2003]
  - Performance, >45 pages
  - Modifiability, >45 pages
  - Security, ~36 pages
  - Availability, ~27 pages
  - Testability, ~22 pages
  - ...and Usability, ~17 pages
Usability in Software Engineering Literature [2]

- The SWEBOK guide has the objective to promote a consistent view of software engineering worldwide and was published 2004 by IEEE.
  - Usability in SWEBOK
    - The word “Usability” used 6 times but usability as such is not at all discussed
    - Software Ergonomics knowledge areas listed in the chapter “Related disciplines of Software Engineering”
  - Performance in SWEBOK
    - The word “Performance” used 32 times and discussed, e.g. performance analysis, performance testing, performance evaluation, performance requirements, performance degradation

Usability and Software Architecture Research Questions

- RQ1: Has usability impact on Software Design?
  - Concern: Is it necessary and possible to teach software engineers usability techniques as well as performance techniques?
- RQ2: How large is the impact of usability on Software Design?
  - Concern: Late usability testing leads to costly revision of software design
- RQ3: Can we build in support for usability in the Software Design early?
  - Concern: Prevent last and costly revision of software design
Usability and Software Architecture Research

- Research groups at the time for the ABB project start-up
  - Usability Supporting Architecture Pattern (USAP), CMU/SEI
    - Bonnie E. John, Len Bass, Elspeth Golden
    - STATUS, European Union project 2001-2003
      - Natalia Juristo, Eelke Folmer, Jan Bosch …

USAP Method

Usability Supporting Architecture Pattern Method
USAP [1]: Study the List of General Usability Scenarios

- Warning/status/alert feedback
  - Status or alert indicators should be displayed when the appropriate conditions are satisfied.

- User profile
  - Each user should be able to set various parameters that control the presentation.

- Command aggregation
  - The user should be able to invoke a file with a collection of commands in it and have the individual commands executed.

- Recovering from failure
  - When the system, the processor, or the network fails, the user should not lose any work in progress.

- Supporting international use
  - The user should have the ability to use the system in a language and with a display format that is familiar to them.

- … and 14 more so far

USAP [2]: Choose Your User Interface’s Scenarios

- Comprehensive searching
- Action on multiple objects
- Reuse of information
- Help
- Identity management
- User Profile
USAP [3]: Study Your Scenarios’ Responsibilities

- Scenario: User Profile
  - Responsibility Category: Authoring
    - Responsibility Sub-Category: Create a Specification
      - AU1.1 The system must provide a way for an authorized author to create a [User Profile]
      - ...
      - ...
  - Responsibility Category: Execution
    - Responsibility Sub-Category: Access the created Specification
      - ...
  - Responsibility Category: Logging
    - Responsibility Sub-Category: Specify the items to be logged
      - ...
  - Responsibility Category: Authorization
    - Responsibility Sub-Category: Authentication
      - ...

USAP [4]: Study Your Responsibilities’ Implementation

- Scenario: User Profile
  - Responsibility Category: Authoring
    - Responsibility Sub-Category: Create a Specification
      - AU1.1 The system must provide a way for an authorized author to create a [User Profile]

Implementation Details for AU1.1.
- The portion of the system that renders output must render a UI that allows the parameters to be specified and displays existing values.
- The portion of the system that accepts input from the user must accept parameters.
- There must be a portion of the system with a mechanism to create a new [User Profile].
USAP [5]: Revise Your System Design

Implementation Details for AU1.1.
- The portion of the system that renders output must render a UI that allows the parameters to be specified and displays existing values.
- The portion of the system that accepts input from the user must accept parameters.
- There must be a portion of the system with a mechanism to create a new [User Profile].

USAP summary

- Describing required system support for user interaction that increases the usability of the scenario/user interface
- Describing implementation details as “portion of the system” implementations for each responsibility
- Reusable USAP artifacts!

- Environment Configuration
- User Profile
- Alarm & Event

Responsibilities

Implementation Details

System Design

User Interface

Usability Scenarios

supports

has many

has many

impact
USAP responses at ABB of 6 hours interaction with USAP tool

- **U1:** We have discussed lots of internal stuff in the system but this gave us some picture of what the user is going to see.

- **U2:** And that is things that we were not going to get that input until very late in the design process, if we hadn’t used this tool now. So it was good to have these points of view come in this early…. I think we have identified at least a couple of new subsystems.

- **U1:** Yes. And some shortcomings of the previous design.

- **U2:** Yeah.

- **U3:** …it was easy to use, of course, no fancy stuff …and one thing that was good was that I need to provide an answer to every question, and in order to provide an answer I need to read text, and it forces me to read the text, so I won’t be able to just skip through it, all the stuff, that was very good.
USAP result at ABB of 6 hours interaction with USAP tool

- SystemABC
  - 14 major issues discovered
    - 6 to be fixed according to guidance in USAP
    - 4 would have been discovered anyway
    - 4 are still open as to whether they will be fixed.
    - Estimated cost savings > 5 person weeks
  - SystemDEF
    - discovered 1 new issue

Finally

Thank you Len, Bonnie and Elsa for a successful cooperation!