

line at any fixed point in time [Kru01].

Variation management encompasses the activities of explicitly representing variability in software artifacts throughout the lifecycle, manages dependences among different variabilities and supports the instantiation of variants. Figure 2.12 shows variation management structured around a product line. In this case, only the common and variant artifacts and the instantiation infrastructure belong to variation management. The product instances are out of the scope from the variation management and are used directly without any further modification via application engineering. Domain engineering deals with the establishment of the platform and product instantiation refers to deriving the products. Product instantiation refers to application engineering.

At the beginning of the present section was mentioned that SPL deals with management over time and space. SPL uses different mechanisms for this kind of variation management. Versioning and branching mechanisms are used for time-based variation and variation point management deals with domain space. Following, a brief overview on these mechanisms is provided:

Version management. Sequential versions of files must be managed over time by operations like Check-in and Check-out.

Branch management. Independent branches of file evolution must be supported by creating parallel file branches.

Variation point management. Points of variation in the software artifacts of the production line must be managed by implementing the variation points in such way, that these include a collection of variants for selecting in the domain space.

2.5. Paper Prototyping

Paper prototyping is a group creativity technique developed to generate different ideas when designing, creating, testing and communicating any type of user interfaces (UIs). The resulting UIs are, later on, used for the selection of an appropriate solution. Examples of such UIs are software programs, applications from mobile devices and websites. Here is a formal definition of paper prototyping:

"Paper prototyping is a variation of usability testing where representative users perform realistic tasks by interacting with a paper version of the interface that is manipulated by a person 'playing computer', who does not explain how the interface is intended to work." [Sny03]

The mechanism of paper prototyping starts with a meeting from all the members of the product team to choose the type of user who represents the target audience for the UI. Afterwards, a *usage profile* is created. Usage profile is a set of typical tasks describing uses for the system. The next step is to make screen shots and/or hand-sketched versions of all the components from the UI required for the usage profile.



Figure 2.13.: Material used during paper prototyping

It is not necessary to have a working version of the interface, since one of the team members will play the role of the computer during the testing.

Figure 2.13 shows the material developed for a hand-drawn paper prototype application used on the laboratory from User-Oriented System Design and Personalized Information Services [Opp09]. The aim of the project, where the author had the opportunity to collaborate, was the development of an application running on a mobile device. The application had to provide with information to find restaurants or bars nearby a proximity radius of 0.5km from the user of the mobile device. The programming language used to develop the application was learned during the lifecycle of the project. The priority of the project was not a final running version but UI design and identification of personalization needs. For this reason design and usability testing were conducted at an early stage by meanings of paper prototyping.

As it has been shown on the previous scenario, paper prototyping enables developers to tests product interfaces before they write code or development begins. This allows for easy and inexpensive modification to existing designs which makes this method useful in the early phases of design. Using paper prototyping allows the entire team to be involved in the process, which eliminates the chance of someone with key information not be involved in the design process, just because he does not have development skills. An additional benefit of paper prototyping is that users feel more comfortable being critical of the prototype because it doesn't have a polished look.

Some areas of application where paper prototyping has been proved to be useful are:

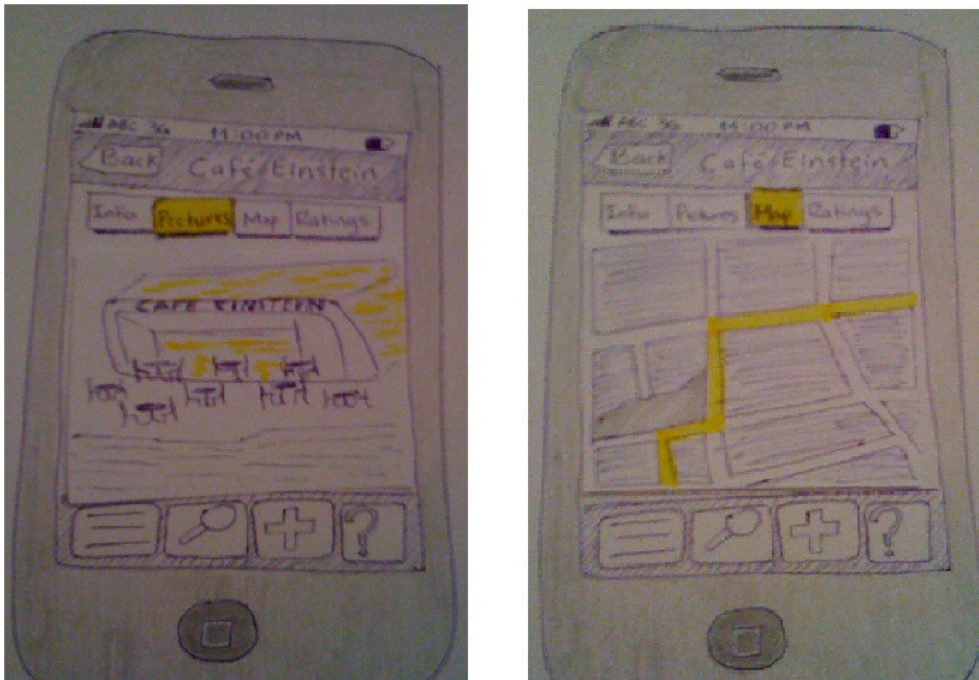


Figure 2.14.: A hand-drawn paper prototype of the screen from a mobile phone application used to search restaurants nearby the user

Communication in the Team. The development team uses paper prototyping to abstract the look of user interfaces (UIs) from an application. Once those UIs are built up meeting the expectations from all the team members, use cases can be used to prove applicability. The paper prototype resulting from the brainstorming can be used as visual specification of the software.

Usability Testing. In such a test, the user performs realistic tasks by interacting with the paper prototype. The paper prototype is manipulated by another person, who's representing the software's response to the user inputs. The method has demonstrated to be successful by identifying usability issues in the design process. There are three techniques of paper prototyping for this kind of testing: compositions, wireframes and storyboards. Compositions deal with the visual aspect, wireframes with the page layout and storyboards are a series of images demonstrating how an interface works. Figure 2.14 depicts a hand-made paper prototype of a mobile phone application. There are 4 buttons at the top of the screen and they are mutual exclusive, which means only one button can be enabled at a time. The highlighted button (color yellow) indicates which screen is currently displayed.

Design Testing. This testing focuses in the easy adaptation from the user to the context of the design. The paper prototype is presented to a user and he is asked with relevant issues such as to identify the main menu or clickable elements.

Rapid Prototyping. This area involves a certain number of designers, each of them creates a paper prototype and test it with a single user. Afterwards, the designers meet

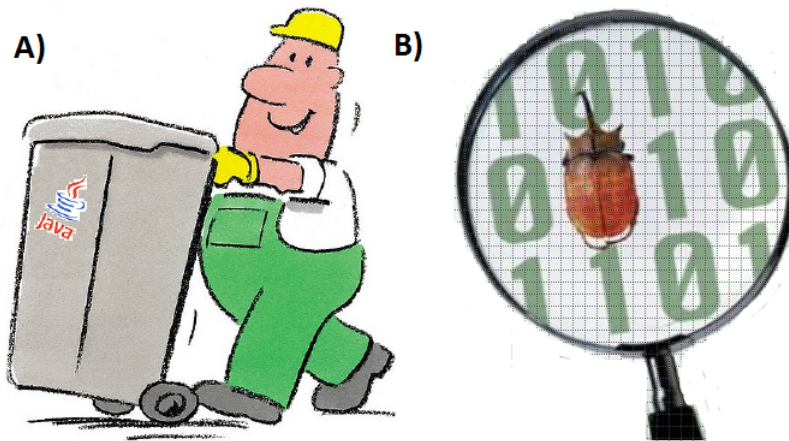


Figure 2.15.: Metaphors in computer science: A) Garbage Collection and B) Debugging

again and discuss the feedback and ideas. The next step is to create a computer-based prototype, where the look and feel is closer to the final product. Nevertheless, the functionality is not an issue during this time. The processes of testing and sharing feedback are repeated and software prototypes are obtained as a result.

According to the four methods previously described, paper prototyping offers benefits regarding the communication within the development team and the quality of the product to be developed. Paper prototypes assure the overall quality of software since they can serve as visual specification of the graphical user interface (GUI). Also, paper prototyping allows communication between the team members regarding the complete design of the required UI. Testing prototypes promote early identification of usability problems even before any code is written, which consequently reduces annoyances of later changes. Finally, but not less important, the costs can be reduced.

2.6. Metaphor in Computer Science

Metaphor is one important writing mechanism which establishes a total identity between two concepts or thoughts without binding them and it is used as symbolic connector. Merriam-Webster's Dictionary defines the word as:

met·a·phor *noun*

I : a figure of speech in which a word or phrase literally denoting one kind of object or idea is used in place of another to suggest a likeness or analogy between them (as in drowning in money); broadly : figurative language - compare simile