

Rethinking the Software Life Cycle: About the Interlace of Different Design and Development Activities

Position Paper for the Dagstuhl Seminar 07081
End User Software Engineering.

Yvonne Dittich

IT University of Copenhagen
Software Development Group
Rued Langgaardsvej 7
DK 2300 Copenhagen, Denmark
+45 7218 5177
ydi@itu.dk

Software engineering research addresses professional ways of designing, developing and implementing software. So far, software engineering more or less takes for granted that software professionals have control over the material implementation of a piece of software. Though users might use the software innovatively or even customise it, neither end-user tailoring (EUT) nor end-user development (EUD) are treated systematically regarding the impact of deferring part of the design to the use context on software development technologies or processes. Especially the development, adaptation and configuration of software products, software that is used by more than one user in more than one organisation – makes visible that different parallel ongoing development activities often distributed over more than two organisations have to be coordinated. To illustrate and develop related issues, let me first present three different projects I was and am involved in:

PD in the Wild

As part of a project focusing on ‘shop floor IT management’¹, we addressed the interlacing of the user side integration and adaptation of software and flexible software processes as part of developing an adequate infrastructure for one-stop service provision. [1] The possibility to tailor system interfaces to allow for exchanging data between different programs and the adjustment the program a developing practice were central for the ongoing design-in-use.

The most appreciated software provider was a small company developing and maintaining a booking system mainly meant for sports facilities. Besides being configured to the specific facilities a municipality provided, the program contained a module which could be tailored to generate ASCII files transferring data e.g. for different economy systems or number-code based access control. The company implemented an agile development process in order to react quickly on bug reports and change proposals. For example the need for an extra field in the customer data to add a mobile number alongside with the landline could be implemented without delay. [2]

¹ *DitA* Design of IT in Use – supportive technologies for public service provision, funded by VINNOVA 2000–2002 at Blekinge Institute of Technology.

Designing for Change

The second project addresses the design and development of tailorable systems for changing business practices at a telecommunication provider.² The main results with respect to this article are that the design of flexible and tailorable systems does not only depend on the requirements from the use context but also the technical environment and the organisation of software development respectively the interaction of these domains. [3] Functional requirements for a tailoring interface might be traded against requirements for maintainability depending on whether the software is developed in-house and small maintenance tasks therefore are less problematic.

Tailoring and end-user development that expands beyond adaptation of applications to individual preferences and - like in this case - affects the model of the common work object implemented by the software requires additional features for designing, testing and debugging have to be provided for the end-user. [4] But even if that can be done, when the limits of the variability provided for the users is reached, software engineers have to take over in order to evolve the program. [5] And here again the connection between use and development plays an important role.

*Design of Evolvable Software Productions*³

The software that provides the cases for this recently started project are ERP systems and simulation software for hydrodynamic system. [6]

Before the software can be applied in a specific organisation or to simulate a specific river system, the software has to be configured, partly by providing data – about the organisation respectively the river system – and partly by configuring the software. In many cases even adaptation to individual preferences is possible. Most of the existing ERP systems can

² The research described here was first funded as a single project *Design in Use of Database Applications* (2000 – 2002). The cooperation then continued as part of *BESQ – Blekinge Engineering Software Qualities*, a profile project (2002-2008) funded by the same agency: The Knowledge Foundation (KKS) at Blekinge Institute of Technology.

³ *ESP – Designing Evolvable Software Products*, funded the Danish government through NABIIT, a research council for nano-, bio- and information technologies from 2006-2008.

also be adapted through a programming interface. Adaptation are often done by independent software houses but can also be implemented by super-user in the end-user organisation.

The different activities of configuration, tailoring and adaptation often implemented using different technical solutions are all contributing to finalising design of the software. And – as could be expected – evolution takes place on all different levels, posing a number of technical and social coordination problems.

Based on the experience from the above-described projects a number of aspects of end-user development become visible as topics for research:

Software Engineering for EU developers

End-user tailoring and development is not only about adjusting personal performance support to individual preferences, but often needed to adjust the existing software to developing work and business processes. This requires to design, implement and test changes in coordination with other users or even with a whole organisation. There is some research on the organisational end-user development, but regarding tools and methods for user-developers more research is needed.

Cooperating across different development sites

There is neither *the* user nor *the* developer. Different members of the use organisation take on different responsibilities regarding the continuing development and adaptation of the software infrastructure. And on the software provider side, development of the base system, configuration and adaptation are often distributed between different organisations. How can the development in the different contexts be organised so that the parallel development processes in the other contexts can be taken into account?

This issue becomes visible when addressing the upgrade of software products. In different projects, we have observed periods of a few weeks, three to four times per year, once a year and several years. What different ways to organise software development correspond to these different upgrade periods. What does that imply for the tailoring and EUD practices of the different products? How can experiences of tailors and EU developers inform the evolution of the base product?

Technical coordination of layered of development

Evolution takes place in parallel on the different layers of a software product. For the different layers, different technologies are used: part of the configuration might result in selection between precompiled alternatives. Other configuration might be saved as meta-data to be interpreted at run-time. The application itself can be changed through a programming language interface. The challenges of this situation become visible when the basic software is updated: base-data, production data, and configuration can be

transformed semi-automatically but might have to be complemented depending on new functionality; adaptations often have to be reworked from scratch or perhaps can now be replaced through tailoring. Every software product provides examples of how to combine different technologies to provide different layers of adaptation and end-user development.

Product line architecture and variability management addresses some of the design issues but mainly in contexts where the deferred design is resolved within the development organisation. There is no systematic categorisation of different possibilities and their combination for multi-layered and end-user development.

Representations of Variability

How to represent complex configuration and adaptation possibilities respectively their constraints so that consultants and end-users can develop a qualified co-design of business organisation and software?

ACKNOWLEDGEMENTS

Thanks to my colleagues Sara Eriksén, Jeanette Eriksson, Christina Hansson and Olle Lindeberg from Blekinge Institute of Technology and Hataichanok Unphon, Peter Sestoft and Sebastien Vaucouleur from the IT University of Copenhagen who have been or are participating in the research projects, this position paper is based on.

REFERENCES

- [1] Y. Dittrich, S. Eriksén, C. Hansson PD in the Wild: Evolving Practices of Design in Use. In: T. Binder, J. Gregory, I. Wagner *Proceedings of the PDC 2002*, Malmö, Sweden.
- [2] C. Hansson, Y. Dittrich, D. Randall How to Include Users in the Development of Off-the-Shelf Software: A Case for Complementing Participatory Design with Agile Development. In: *Proceedings of the HICSS 2006*.
- [3] Y. Dittrich, O. Lindeberg Designing for Changing Work and Business Practices. To be published in: N. Patel (ed.) *Evolutionary and Adaptive Information Systems*, Idea Group Publishing: 2002.
- [4] J. Eriksson, Y. Dittrich Combining Tailoring and Evolutionary Software Development for Rapidly Changing Business Systems. Accepted for the *Journal of Organizational and End-User Computing*.
- [5] Y. Dittrich, O. Lindeberg, L. Lundberg End- User Development as Adaptive Maintenance. Two Industrial Cases. In: V. Wulf, H. Lieberman, F. Paternó (eds.) *End User Development: Empowering people to flexibly employ advanced information and communication technology*, Springer 2006.
- [6] ESP web address