Larisa GAVRILA
Universitatea Politehnica Bucuresti
George BASICA
Universitatea Petrosani
Sorin IONESCU
Universitatea Politehnica Bucuresti

SOFTWARE PRODUCTIZATION

Case Study

Keywords
Software productization,
Software personalization,
Prototyping

JEL Classification
M15

Abstract

The purpose of this article is to define software personalization and software product. As on the market there are several types of software they all have in common the fact that they need personalization, whether that is possible or not in some cases. The idea is to have a solid backbone/core that can support building on top of it tailored functionalities. That backbone can also be a prototype and then the question that emerges is related to the characteristics that a prototype should have in order to turn into a software product that can be personalized. Authors have researched this aspect and present the results of their study in this article, this being actually the novelty that this article brings up.

LITERATURE REVIEW AND CONCEPT DEFINITIONS

In the specialized literature software is defined as: "a package of configured components or a software-based service, with ancillary materials, which is launched and traded in a specific market" (Xu and Brinkkemper, 2005). Also researchers have identified (Peter et al, 2010; Sawyer, 2000; Natt, 2005; Alves and Castro, 2006) a number of types of software: packaged software, bespoke software, "off the shelf" software.

The packaged software is defined as software sold as a negotiable commodity/product (purchased from a vendor, distributor or store) for all computers/platforms, workstations, usually packaged software is licensed for use, not sold (Peter et al, 2010; Sawyer, 2000). Bespoke software is also known as software development on the basis of a contract with a specific client (Peter et al, 2010; Natt, 2005). Software off the shelf is driven by market demands and applications offer many more functions compared to the clients' needs (Peter et al, 2010; Alves and Castro, 2006).

Moving forward, it will be defined the notion of software product as it is seen and accepted by the authors in this article as well as the notion of software personalization.

The software represents a mix of informatics components and their associated services that together bring added value to the user through the use of the product in question. In Figure 1 you can see parts of a software product as defined in the author's view:

The core contains the basic product functionalities that allow the application to function and to be freestanding. The starting point for the core may actually be a prototype. The term productization does not officially exist in English language nor in the dictionary although it is increasingly common and mentioned in the technical literature. The term can be defined as a standardized process that aims to produce from available information a good high quality commercial good or service viable in the market (Arho et al, 2009). In another article, productization consists of analyzing the needs of the customers in the target market, product design and capabilities development to produce the product (Flamholtz and Aksehirli, 2000).

The productization stage is defined by producing goods which may be placed on the market and the focus is on documentation, training and transfer to the production environment, the duration of this phase is variable depending on the size of the product (O'Neill et al, 1990). The main objective of productization is, therefore to use and re-use the know-how that has been gained from previous projects (Ojanen et al, 2007). Productization means to standardize the items to offer, the term productization includes many technological elements from the very early stages of designing a product (i.e., requirements management, selection of technology platforms, designing product architecture etc.) to commercial sales and product distribution (i.e., delivery channels, market positioning and product sales activities) (Hietala et al, 2004).

This article refers to prototype productization, which involves the repetition of several cycles of defining requirements, coding, testing and system development, until the moment the prototype becomes stable, reliable and standardized so that it can be transferred to core level that is self-contained and fully functional (as you can see in Figure 2).

STUDY METHODOLOGY

The results of two studies are presented in this article. The first study is only a calibration study that had the main purpose to identify the key characteristics that a prototype should have in order to be personalized later on. Being a calibration study only, the authors have targeted only 26 respondents. After the results of the first study were identifies authors have pursued further and expanded the group target. The main purpose of the second study was to identify what are the most important characteristics from the ones mentioned in the calibration study. The second study also pays more attention and makes a distinct difference between the nature of the respondents (business or technical) in the attempt to identify how characteristics are perceived by two distinct groups. The study was conducted for one month in July 2017 and the second study was conducted also for one month in October 2017. The method used is crowdsourcing. In Appendix 1 are mentioned the distribution channels for this study.

STUDY RESULTS

It should be underlined that the prototype moved to kernel-level must be able to withstand changes in order to add new features or in order to be personalized. Such a prototype must have certain characteristics in order to constitute the nucleus of a software product and to be customizable in the future.

As demonstrated in Figure 3,the main characteristic of a prototype that can support personalization are:

- flexibility (32%)
- generic backbone (16%)
- incorporated user experience (10%)
- modularity (10%)
- availability (7%)

- documented (6%)
- extensible (3%)
- scalable (3%)

The flexibility character, in the context of this study, is given by how the prototype is built from the perspective of the code that determines how adaptable, flexible the prototype in question becomes, also the prototype should be configurable and to have accessible programming interfaces so that the prototype can be customized without modifying the source code.

The generic character is given by the incorporation of general work flows and an advanced grouping feature. In terms of user experience, this should also be ensured by using a friendly interface. All at once the prototype should be made up of a series of modules that can be added or erased relatively easily. To win a competitive advantage, the prototype should be available on the market in a very short time. An important aspect is the associated documentation of the prototype that is need in order to be able to track exactly what has been customized and how does the architecture work in the event of incidents.

When analyzing the data from the extended results (see Figure 4) where 100 professionals have responded to the survey, it is more clear that the top 3 characteristics of a prototype to support personalization are: to be configurable, flexible and scalable. This is the view both for those having business roles as well as for those have technical roles.

A good application programming interface, flexibility and scalability are definitely more important for technical roles than business roles. Almost the same levels of importance for the two roles have characteristics like generic modules and configurability. User friendly characteristic is more important for business roles than technical roles.

Once the core is functional, the extra functionalities around it can be added to refine the initial application, either to extend the area of use. The core and extra features make up the informational components of the software. Around the informational components, their associated services can be built like support, maintenance, training and consultancy.

In this article software personalization is defined as any change made to the software product, it includes modifications to source code (customization), to parameters (configuration) or modifications made to the associated services.

CONCLUSIONS

In the specialized literature software services and products are defined in different ways, in this artcile the authors have defined the software as being a mix of informatics components and their associated services that together bring added value to the user through the use of the product in question.

This article refers to prototype productization, which involves the repetition of several cycles of defining requirements, coding, testing and system development, until the moment the prototype becomes stable, reliable and standardized so that it can be transferred to core level that is self-contained and fully functional.

Two studies have been conducted on more than 100 professionals to identify the most important characteristics of a prototype that can support additional features/functionalities with the aim of personalization. After the extended study was executed, the conclusion is that it is more clear that the top 3 characteristics of a prototype to support personalization are: to be configurable, flexible and scalable. This is the view both for those having business roles as well as for those have technical roles.

REFERENCES

[1] Xu, L., & Brinkkemper, S. (2005). Concepts of product software: Paving the road for urgently needed research, Proceedings of the 1st International Workshop on Philosophical Foundations of Information Systems Engineering (LNCS), Springer: Berlin, pp. 523-528.

[2] Peter Artz, Inge van de Weerd, Sjaak Brinkkemper (2010), Productization: The process of transforming from customer-specific software development to product software development, Technical Report UU-CS-2010-003 January 2010, Department of Information and Computing Sciences Utrecht University, Utrecht, The Netherlands, ISSN: 0924-3275, pag 16.

[3] Sawyer, S. (2000). Packaged software: implications of the differences from custom approaches to software development, European Journal of Information Systems, 9(1), pp. 47-58.

[4] Natt och Dag, J. (2005). Managing Natural Language Requirements in LargeScale Software Development, ISSN 1101-3931, ISRN LUTEDX/TETS–1070–SE+222P. Department of Communication Systems. Sweden: Lund University.

[5] Alves, C., & Castro, J. (2006). A Study in Market-Driven Requirements Engineering. 9th Workshop of Requirements Engineering.

[6] Arho Suominen, Jussi Kantola, Aulis Tuominen (2009), Reviewing and Defining Productization, Research Gate (https://www.researchgate.net/publication/236326445_Reviewing_and_Defining_Productizat

ion?enrichId=rgreq-e60ce8a0ad833fe82634525d4c60cd4f-XXX&enrichSource=Y292ZXJQYWdlOzIzNjMyNjQ0NTtBUzoyOTk5NTY1NjE4MjU3OTNAMTQ0ODUyNjYyNTIyMw%3D%3D&el=1_x_2&_esc=publicationCoverPdf), accessed 28.08.2017

[7] Flamholtz, E.; Aksehirli, Z. (2000), Organizational success and failure: an empirical test of a holistic model,; European Management Journal Volume 18, Issue 5, Pages 488-498

[8] O'Neill, D.M.; Mullarkey, P.W.; Gingrich, P.C. (1990); The LQMS research project: a case study in knowledge-based system development; Managing Expert System Programs and Projects, 1990. Proceedings., IEEE Conference on 10-12 Sept. 1990, Page(s):61 – 69

[9] Ojanen V., Salmi P., Torkkeli M. (2007), Innovation Patterns in KIBS Organizations: A Case Study of Finnish Technical Engineering Industry; System Sciences; 40th Annual Hawaii International Conference on Jan. 2007, Page(s):228b - 228b

[10] Hietala J.;Kontio, J.; Jokinen, J-P.; Pyysiäinen, J. (2004), Challenges of Software Product Companies: Results of a National Survey in Finland, ; 10th IEEE International Symposium on Software Metrics (METRICS'04), Page(s) 232-243.

Appendices

- ProjectManagement.Com
(https://www.projectmanagement.com/discussion-topic/52524/Software-customization-projects)
-The Project Manager Network - #1 Group for Project Managers
(https://www.linkedin.com/groups/37888/37888-6233951198957961216)
- Telecoms Professionals: IoT, LTE, M2M, OTT, Internet of Things, Mobile, Telecom
(https://www.linkedin.com/groups/23013/23013-6233949068633210882)
- Project Manager Community - Best Group for Project Management
(https://www.linkedin.com/groups/35313/35313-6231031368529383427)
- PMLink - Project Management Link - Project, Program & Portfolio Managers, PMP, PMBOK, PMO
(https://www.linkedin.com/groups/59531/59531-6231030086091575299)
- Big Data, Analytics, Business Intelligence & Visualization Experts Community
(https://www.linkedin.com/groups/23006/23006-6233948992544350210)
- Telecom, Mobile Apps & Payment Professionals Worldwide
(https://www.linkedin.com/groups/90498/90498-6233950370050244610)
- PMO - Project Management Office
(https://www.linkedin.com/groups/80342/80342-6229737207725457412)
- I want to be a PMP®
(https://www.linkedin.com/groups/2356441/2356441-6231032452828274688)
- Project Management Group SP
(https://www.linkedin.com/groups/47251/47251-6231031887327035395)
- Global Project Management
(https://www.linkedin.com/groups/2775/2775-6229737655308029957)

Appendix 1, Channels used to distribute the study survey

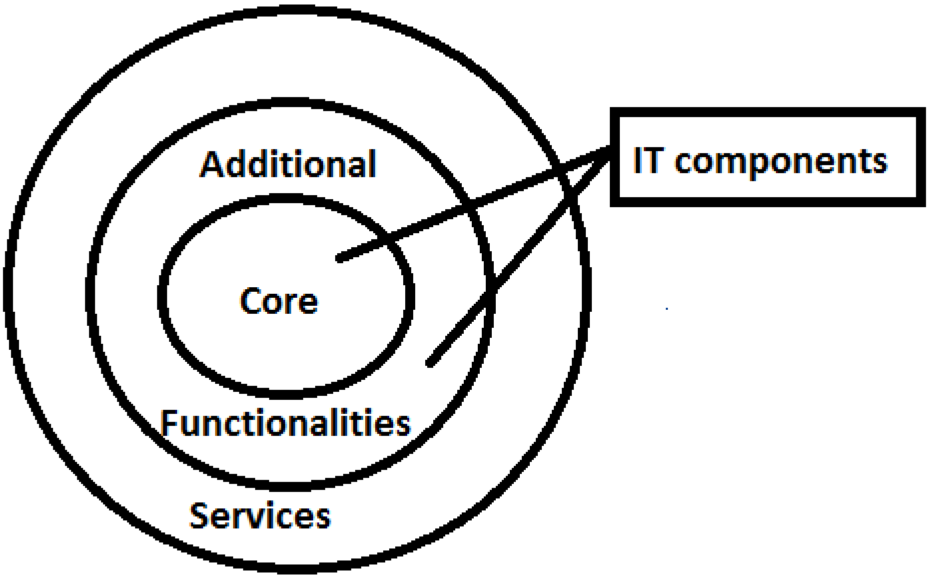

Figure 1. Software product

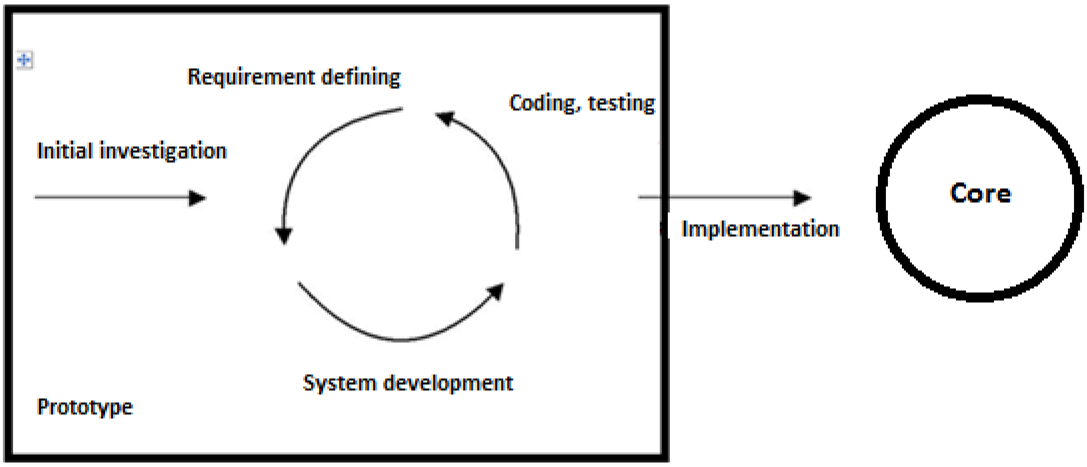

Figure 2. Prototype productization

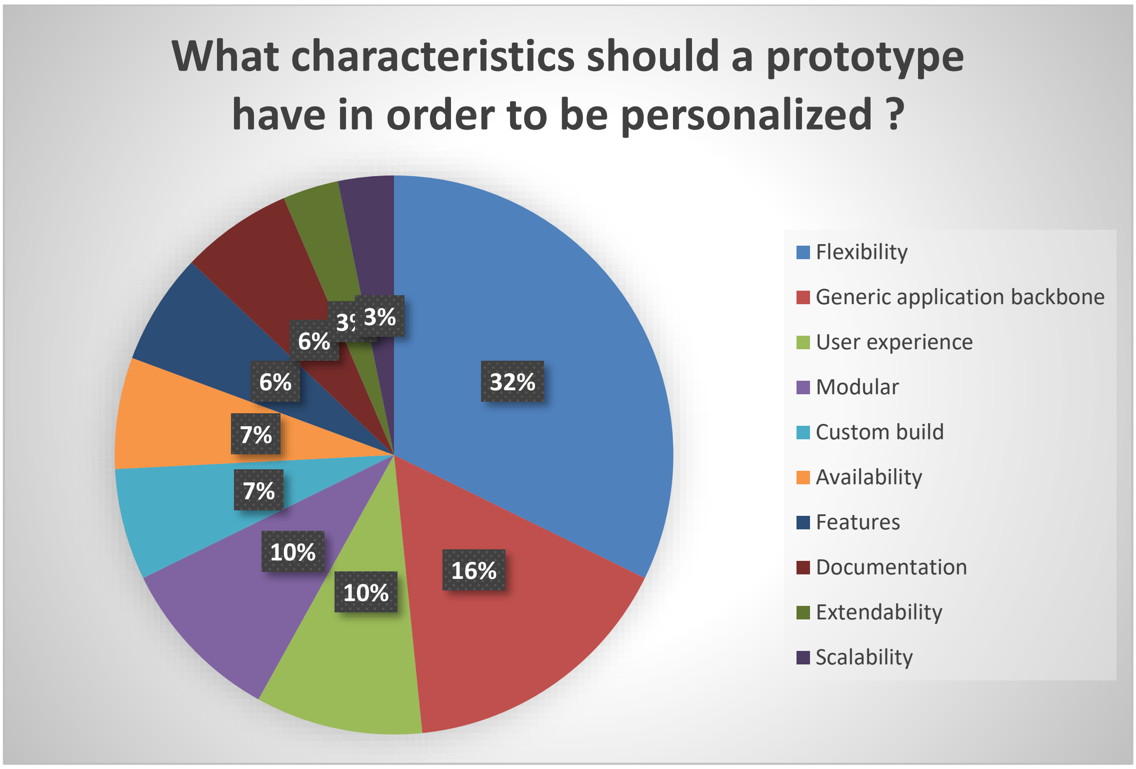

Figure 3. Calibration study results

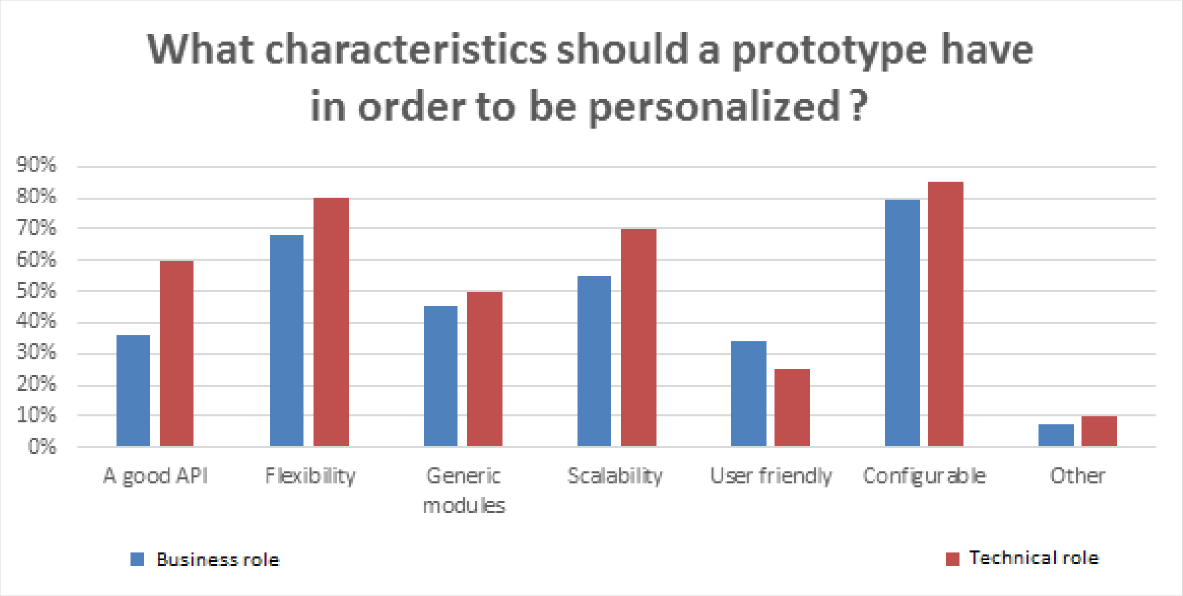

Figure 4. Extended study results