Automatic Reuse of Prototypes in Software Engineering: A Survey of Available Tools

A. Sánchez-Villarín[1] [a], A. Santos-Montaño[1] [b] and J. G. Enríquez[2] [c]

[1]*IWT2 Research Group, University of Seville, Spain*

[2]*Computer Languages and Systems Department, University of Seville, E.T.S. Ingeniría Inoformática,*
avda. Reina Mercedes s/n, 41012, Seville, Spain

Keywords: Prototypes, Prototypes Tools, Prototypes Reuse.

Abstract: The use of prototypes as an excellent mechanism for communication between software users has been fully accepted in the literature. Both the academic and business worlds agree on its use as a software technique, used primarily to capture requirements and as a means of communication with the user. However, often, prototypes are developed very quickly or without the collaboration of the users. This is one of the main reasons why the real power of the prototypes is not used in software development. The initial hypothesis of this research is that this problem occurs because prototypes are regarded as disposable milestones. This paper analyzes whether there are adequate tools on the market to help development teams to reuse prototypes as an exceedingly mechanism for starting and expediting new software development projects. With the present study we show different tools for prototypes that, even though they are a preliminary evaluation, show the disadvantages to be solved in the future development and research of solutions.

1 INTRODUCTION

The use of prototypes to improve the communication with clients are final users is a technique not only considered in software engineering. In fact, it is a technique considered in mainly any product development. Thus, for instance, the design thinking methodology (Brown et al., 2008; Dym et al., 2005) proposes the definition of prototypes as the base for each product definition and it has to be developed after empathizing, defining and thinking with final users.

In software engineering context, interface prototypes are recognized as a very powerful tool for the communication with client and final users (Alavi, 1984). However, in practice, it has been noticed that software and interfaces prototypes are developed in a very fast way, with very few resources, and even, without the enrollment of users. Results of this process normally are discarded in the rest of the life cycle and it provokes that some software teams and companies consider them as a "required play" that only use resources with a minimum of benefices (Escalona

---

[a] https://orcid.org/0000-0002-7740-5327
[b] https://orcid.org/0000-0002-0749-6411
[c] https://orcid.org/0000-0002-2631-5890

et al., 2004).

Under our experience with companies (Escalona and Koch, 2004), this idea is not so wild. When we consider to develop prototypes in the development life cycle (mainly in the early phases), we have to decide several critical aspects like use high or low confidence prototypes or the use of vertical or horizontal prototypes among others. These decisions require time and resources and if the prototypes are considered as a disposable milestone, this cost seems not to be justified. Another option to this situation is to try to develop prototypes that could be "reused" in the next activities of the software development life cycle, for instance, in interface design. However, translating prototypes to software interfaces requires a manual process, consequently, it produces an increase of cost again.

To find a solution to this situation, there are several tools in the market that promise to help to the development teams to develop prototypes that could be "automatically" translated into future software interface. Thus, the investment of prototypes development is justified and the team could benefit from prototypes advantages.

In this paper, we present a global view about tools that promise to help us in the process to generate analysis and design products or even coding prototypes

---

www.manaraa.com

automatically or, at least, semi-automatically.

The rest of this paper is structured as follows. Firstly, we starting with a very short introduction of the context where this research is being developed, which is very closed to the enterprise world. In Section 3, the methodology followed for this study is described. Section 4 instances the methodology proposed in previous section and presents the results obtained. Section 5 presents learned lessons and finally, paper finishes with a set of conclusions and future works.

2 CONTEXT

Coding prototypes with the same precision with which they were designed is a real problem in daily programming (McMillan et al., 2012). Unfortunately, current methods found in both literature and industry do not allow prototypes to be transformed into modifiable code, following requirements and characteristics that must be met. In a Scrum (Schwaber and Beedle, 2002) or Design Sprint methodology (Banfield et al., 2015) context, the importance of having validated prototypes saves a lot of time and avoids future problems if the programming is in an advanced state.

The problem that is cited is common within the iMedea project (innovative Medical Engineering Assistance) (Escalona, 2019) by G7Innovation (G7 Innovation, 2019) in Seville, in which the prototypes are validated in collaboration with the Inebir clinic, but there are details or formulas that escape once they are translated to the code. G7 Innovation is a SME (Small and Medium Enterprise) focused in the development of software oriented to sanitary environments. In fact, they produce high TIC results in the field of Smart Laboratories, Human Reproduction or specific results for chronic treatment using different devices and sensors (V. Cid, 2019; L. Morales, 2019). In this context, the communication between the sanitary environment and the IT engineers is a critical factor to assure the quality of the results. For this reason, they used they have developed, a version of a methodology named NDT (Navigational Development Techniques) (Escalona and Aragón, 2008) that, in its current version mix Scrum with Design Thinking principles.

It is not the aim of the paper the presentation of the methodology but, basically, it covers three main steps: the first one is to "discover the problem". In this phase, users and engineers work together to define the problem and a first set of prototypes for the solution. These prototypes are improved in different iterations and, at the end of this first step they get a very high-fidelity prototype. After that, in the second

step, the IT Team execute the life cycle to analyze, design and built the software to support these prototypes. In the third steps, the result is implanted and deployed in the environment. It is only a short presentation but, obviously, each step has a set of phases, task and techniques to be applied that can be obtained from (NDT-Navitgational Development Techniques, 2019). However, this global view, let us to introduce our motivation to present the problem that we want to try to start to solve with this paper. The team in iMedea requires that, after design a very high-fidelity prototype, they must create the analysis, the screen or any aspect from zero, and the original prototypes are only used for communication. They claim if it will possible to try to reuse them to automatically generate products in the next phases automatically. In fact, NDT use the Model Driven Paradigm (MDE) (Domínguez-Mayo et al., 2012) in other aspects, for instance, to produce test from requirements. So, the question that motives this work is: could be possible to use prototypes defined and validated with the user to generate products in the rest of the life cycle?

In this sense, it is necessary to use tools that optimize the process from obtaining the validation of the prototypes to the already programmed views, evaluating that it is possible to obtain a code that can be modified later and that is faithful to the styles of the prototypes. To this end, it will be inevitable that the prototypes are made with delicacy, effort and enthusiasm, as this would save time in programming.

3 METHODOLOGY

Systematic Literature Reviews (SLRs) (Kitchenham and Brereton, 2013) and Systematic Mapping Studies (SMSs) (Petersen et al., 2008) are some of the most commonly used methods for performing state-of-the-art studies in software engineering. The choice of each of them depends on the objectives to be achieved with the research.

SLRs are more expensive to perform than SMSs (Velthuis, 2014). The main objective of an SLR is to identify best practices based on empirical studies, the one of SMSs, is to provide a classification and a thematic analysis of a concrete topic.

The most widely accepted SLR method around software engineering, is the one proposed by Kitchenham (Kitchenham and Brereton, 2013). This proposal concludes that a review of the literature should consist of the following phases:

- Planning: before an SLR, it is necessary to confirm the need for the **research**. The most important activity is writing the research questions that

define the review protocol.

- Conducting: is about executing the protocol that is defined.

- Reporting: describe how the final report is prepared.

The method that defines the process of a SMS proposed by Petersen et al. (Petersen et al., 2008), shows a lot of similarities with the one presented by Kitchenham, establishing set of five steps, where each of them produces an output. These steps are:

- Definition of the Research Questions: this activity is related to the formulation of the Research Questions (RQs) that will guide the work.

- Conduct Search: this activity is related to execute the search, based on the RQs. The search it is normally executed in different digital libraries and based on some keywords extracted from the RQs.

- Screening of Papers: this activity is related to apply the inclusion and exclusion criteria with the aim of selecting the most relevant and close papers to the topic of the research.

- Keywording using Abstracts: this activity is related to the building of the classification scheme, where all the primary papers selected in the previous phase will be categorized.

- Data extraction and mapping process: this final activity, is related to the data extraction and the mapping process based on the results obtained in the keywording activity. This activity will let the researchers to classify which is the state of the art of the topic and to identify gaps and possibilities for future research.

SLRs and SMSs are considered valid formal methods for carrying out studies and surveys related to technological tools or solutions, so they were our first consideration to execute our study. However, there are some limitations, such as: defining search engines or keywords, among others, that do not allow to execute any of the methodologies strictly. Considering that the scope of this work is much more focused on the industrial than on the scientific literature, it has been taken as a reference.

4 COMPARATIVE STUDY

At the time of beginning the study of the tools, it was thought to use the search and selection mechanism proposed by Barbara Kitchenham, which is known as SLR. However, thanks to what was contributed in the paper (Escalona et al., 2014), "it was seen that SLRs

proposals are highly directed towards answering research questions on some scientific knowledge. Nevertheless, SLR is not enough for a study led to compare technologies or tools solutions".

Therefore, the study of the tools and the questions of the questionnaire presented below have been obtained in an empirical way analyzing the needs of work groups in projects, as is the example of iMedea project by G7Innovation, which had a clear need to improve for the modelling of the prototypes, as it is introduced in Section 2. Through a follow-up of the project and by means of questions to the members of the team it was possible to obtain the following questionnaire.

On the other hand, tools were obtained empirically analyzing which are the most used in the current market of the software industry and the most popular among the users.

4.1 Characterization Map

According to previous section, our idea is to try to create high detailed and high-fidelity prototypes, which could be automatically translate into analysis, design products or even final screens. For this aim, and focusing our work in iMedea scenario, a research about systems that allow to convert interface prototypes designed into a typed prototype such as XML or HTML has been done. It has been proved many different tools, applications such as Axure or JustinMind, or online converters like the web "zamzar.com". After proving and analyzing all the investigated tools, we realized that no one tool did exactly the same that we were researching, due to the tool did not convert to the format we wanted or because the tool returned conversions very little satisfactory, then it would not be able to be used.

Following this idea, it has been elaborated a series of questions that allow to validate if a tool offers good functionalities to do the conversion of the prototypes. They are formed by a set of trichotomy questions whose affirmative answer indicates that the tool owns a good characteristic, the negative that indicates that it has not that characteristic or an intermediate answer that indicate that it has only that characteristic partly. These questions were developed in the collaboration of iMedea team and they are the next, sorted by relevance:

- **Q01** - Does it allow to convert or export prototypes into multiple formats, such as XML and HTML?

- **Q02** - Is the conversion done correctly, exporting it with the correct tags?

- **Q03** - Is possible to interact with the system via API?

- **Q04** - Also to convert into different formats, does the tool allow to create its own prototypes in HTML or another typed language?

- **Q05** - Are there versions of the tool for the main operative system such as Windows, Mac OS and Linux?

- **Q06** - Is the complete system free?

- **Q07** - In the case that the tool is paid, are there licenses for students and universities?

- **Q08** - Does the tool receive periodically updates and is followed by a community?

- **Q09** - Does the tool have a good customer service?

- **Q10** - Is the system intuitive and user friendly?

In order to answer these questions, we defined several searching criteria in order to look in the most relevant research search engines: Scopus, Scholar, etc. but, as we introduce, this engines are not the most suitable for tools searching and we had to move to google following criteria defined in (Escalona et al., 2014). Thus, we obtained a set of ten suitable tools that seem cover part of our requirements. Next sections present an analysis of all of them.

4.2 Selected Tools

Below is shown a summary of the study made to the tools used that will be presented, analyzing the pros and cons of each one. After applying several searching criteria, we selected tools presented in Table 1. Each tool is identified by a code and for each of them, a references and short description is presented and also a set of advantages and disadvantages in a very short way.

4.3 Evaluation

A high fidelity prototype defined by iMedea team that was validated with the user, was taken as reference to be redesigned for each tool [1].

Two juniors engineers have modelled this example in each numerated tool getting the results that are presented in Table 2 supervised by two senior researchers. Thus, this table presents an instance of the characterization map presented in Section 4.1.

---

[1]We cannot present the concrete prototype that we used because it is property of G7 Innovation but if any reader is interested in knowing more about the prototype and iMedea, please contact with authors to share information.

In table, we find each question in columns and each tool in rows. Each cell presents the results of the evaluation of each question for each tool. In each cell, three values are possible:

- **Y:** It means "Yes" and represents that this tool completely answers this question.

- **N:** It means "No" and represents that this tool does not support this question at all.

- **P:** It means "Partially" and represents that this tool only supports this question partially.

5 LEARNED LESSONS

From Table 2 we found that none tool is completely suitable for our aim. We analyze in detail.

Q01 - Does it allow to convert or export prototypes into multiple formats, such as XML and HTML?
Only three tools support this requirement. It is something very relevant because we require to get a format that we can analyze and generate analysis source products from them.

Q02 - Is the conversion done correctly, exporting it with the correct tags?
To answer this question, we test the tool with a real prototype for iMedea. We detected that conversion or generation (in the tools that allow some of that) have fails or produce different mistakes.

Q03 - Is possible to interact with the system via Application Programming Interface (API)?
Only two tools offer this possibility to us. The global idea that we have it is to analyze the "core" of our prototypes and to generate analysis and/or design elements from it, for instance, using mechanisms like Model Driven Engineering, but, if we are closed to its "core", the tool must be completely discarded.

Q04 - Also to convert into different formats, does the tool allow to create its own mock-ups in HTML or another typed language?
Different formats are not so relevant for iMedea project because the environment is a fixed one, but, if we think a bigger company with different environments, it can be a very restrictive aspect. In this sense, none tool allows to convert in different formats.

Q05 - Are there versions of the tool for the main operative system such as Windows, Mac OS and Linux?
This question is, like question Q04, more relevant for big companies. As can be deduced from Table 2, tools are focused on a concrete format.

Table 1: Overview of selected tools.

ID	Tool	Description	Advantages	Disadvantages
A01	Balsamiq (Balsamiq, 2019)	Application that facilitates and speed up the creation of sketches, both for web and mobile.	Simplicity.	Finished unprofessional. Payment version.
A02	Microsoft Expression Blend (Microsoft Expression Blend, 2019)	Professional tool of design, with the final purpose to provide experiences of users attractive.	Finished professional.	Payment version.
A03	Adobe XD (Adobe XD, 2019)	Tool based in vectors to design and create prototypes of the user experience for applications.	Finished professional. Free version.	Complex use.
A04	moqups (Moqups, 2019)	Web application simplified and intuitive that helps to create and collaborate in wireframes, models, diagrams and prototypes.	User friendly.	Payment version.
A05	invision (Invision, 2019)	Platform of design of digital products that is used to create the best experiences of clients of the world.	Finished professional. User friendly. Free.	It has not API.
A06	Zeplin (Zeplin, 2019)	Application to use in team, to the design with guides of precise style or code fragments automatically.	Intuitive use.	Payment version.
A07	Zamzar (Zamzar, 2019)	Web application that allows to convert Mockups made in PowerPoints into HTML, XML and other languages.	Easy and intuitive. It has API.	Payment version. Poor conversion.
A08	Wireframe Sketcher (Wireframe Sketcher, 2019)	Tool to create wireframes, models and prototypes for applications, is multiplatform.	Compatibility with Eclipse IDE. Finished unprofessional.	Use few intuitive.
A09	Justinmind (Justinmind, 2019)	Tool to create prototype adapted to mobiles.	Intuitive use.	Payment version.
A10	Axure (Axure, 2019)	Tool to create prototypes and convert them into other languages	Finished professional. Adapted to mobile. Easy and intuitive. It has API.	Payment version.

Q06 - Is the complete system free?

We can deduce that there is no tool is completely free access. It should be not a big problem but, if we analyze the requirements that non-free access tools offer in the rest aspect, it is difficult to justify the investment in a SME company for acquired a tool. This is a relevant question, since the costs of the tools should not be ignored, even if they differ greatly from personal expenses.

Q07 - In the case that the tool is paid, are there licenses for students and universities?

In our environment, this question is quite relevant, perhaps, it is not so relevant for enterprise environ-

Table 2: Characterization map.

	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	Q09	Q10	Totals
A01	P	N	N	N	P	P	N	Y	P	N	Y: 1 N: 5 P: 4
A02	N	N	N	N	N	P	N	N	P	Y	Y: 1 N: 7 P: 2
A03	Y	N	P	N	N	P	N	N	P	Y	Y: 2 N: 5 P: 3
A04	Y	N	P	N	P	P	N	N	P	Y	Y: 2 N: 4 P: 4
A05	P	Y	N	N	P	P	N	N	P	Y	Y: 2 N: 4 P: 4
A06	P	N	P	N	P	P	N	N	P	Y	Y: 1 N: 4 P: 5
A07	Y	P	Y	N	Y	P	P	P	N	Y	Y: 4 N: 2 P: 4
A08	P	P	N	N	N	P	N	N	P	N	Y: 0 N: 6 P: 4
A09	P	N	P	N	P	P	N	N	P	Y	Y: 1 N: 4 P: 5
A10	P	N	Y	P	Y	P	N	P	P	Y	Y: 3 N: 2 P: 5

ment, but universities and students are a very relevant community to test, to improve or even to analyze tools like this one. We can see that very few consider this aspect.

Q08 - Does the tool receive periodically updates and is followed by a community?
To have a relevant community or support is essential in any tool acquisition. From results in Table 2, we deduce that few of tools have a good or a partially good community to support its implantation.

Q09 - Does the tool have a good customer service?
To answer this question, we have navigated by customer client services of the different tools. We have analyzed opinions and, in general, it seems not to be too much critics to the support. However, we consider that this question should be analyzed under a contract in case of acquiring a tool and it depends of the concrete environment where it will be used.

Q10 - Is the system intuitive and user friendly?
To answer this question, we value the environment by ourselves experiences. As we introduced, it is a preliminary study and it was executed in a very concrete environment with few resources (only two junior software engineers as testers). However, in general, we consider that tools are friendly and easy to use. It is relevant in environments like iMedea one, because in companies, mainly in SME, the cost of the learning curve in new tools is a critical aspect to be considered.

Finally, with learned lessons, we consider that, with this very preliminary work, the current situation shows that there is an important gap in software engineering in this aspect. It opens a tool for our future research with the assurance that it is also relevant for companies.

6 CONCLUSIONS AND FUTURE WORKS

This paper has presented a preliminary comparative study about tools for software prototyping. Starting from a real problem detected in our experience with companies, the scenario presented in Section 2, we try to follow Escalona et al. (Escalona et al., 2014) to compare and try to the current situation.

In the context of a very concrete problem, a SME called G7 Innovation, which develops a concrete product based on prototypes called iMedea, we have defined a characterization schema based on ten relevant questions. With a team composed by two junior software engineers, these questions have been answered for each tool.

We are aware that this study has important gaps that we want to try to solve in our future works. First, searching was not exhaustive. After this study, we are considering trying to make a research SLR to try to look, not tools, just approaches or research related with our aims to try to find inspiration about how to try to solve the problem. In fact, we read previous experience, like NDT-Prototypes (García-García et al., 2012) that proposes to use Model-Driven Engineering, it could be a future way to consider.

Other important gap is about the definition of the schema. The definition of the questions was made in the concrete context of iMedea and we must improve them defined research questions that cover a higher spectrum.

Searching criteria is also a very important gap. As we introduce, classical research search engines were not a good alternative because we were looking for tools. Obviously, if we want to execute a SLR, we have not only to define a good set of research question, just also a good set of search criteria.

Despite of these gaps, we can conclude that this preliminary evaluation, opens an important opportu-

nity to research. Obviously, the situation of G7 Innovation and iMedea is very frequently, so the enterprise requires this kind of solutions. With this study, we have put the first seed to start to research and to develop solutions in this direction.

ACKNOWLEDGEMENTS

REFERENCES

Adobe XD (Last Accessed: July, 2019). https://www.adobe.com/es/products/xd.html.

Alavi, M. (1984). An assessment of the prototyping approach to information systems development. *Communications of the ACM*, 27(6):556–563.

Axure (Last Accessed: July, 2019). https://www.axure.com/.

Balsamiq (Last Accessed: July, 2019). https://balsamiq.com/.

Banfield, R., Lombardo, C. T., and Wax, T. (2015). *Design sprint: A practical guidebook for building great digital products*. " O'Reilly Media, Inc.".

Brown, T. et al. (2008). Design thinking. *Harvard business review*, 86(6):84.

Domínguez-Mayo, F., Escalona, M. J., Mejías, M., Ross, M., and Staples, G. (2012). Quality evaluation for model-driven web engineering methodologies. *Information and Software Technology*, 54(11):1265–1282.

Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., and Leifer, L. J. (2005). Engineering design thinking, teaching, and learning. *Journal of engineering education*, 94(1):103–120.

Escalona, M. (2019). imedea: Una solución orientada en el paciente. In *Proceedings of Infors@alud*, page 239.

Escalona, M., Mejías, M., and Torres, J. (2004). Developing systems with ndt & ndt-tool. In *13th International conference on information systems development: methods and tools, theory and practice, Vilna, Lithuania*, pages 149–59.

Escalona, M. J. and Aragón, G. (2008). Ndt. a model-driven approach for web requirements. *IEEE Transactions on software engineering*, 34(3):377–390.

Escalona, M. J., García García, J. A., Dominguez-Mayo, F. J., and Ramos, I. (2014). Technical tool surveys and comparative studies: a systematical approach.

Escalona, M. J. and Koch, N. (2004). Requirements engineering for web applications-a comparative study. *J. Web Eng.*, 2(3):193–212.

G7 Innovation (Last Accessed: July, 2019). http://g7innovation.com/.

García-García, J. A., Ortega, M. A., García-Borgoñon, L., and Escalona, M. J. (2012). Ndt-suite: a model-based suite for the application of ndt. In *International Conference on Web Engineering*, pages 469–472. Springer.

Invision (Last Accessed: July, 2019). https://www.invisionapp.com/.

Justinmind (Last Accessed: July, 2019). https://www.justinmind.com/.

Kitchenham, B. and Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and software technology*, 55(12):2049–2075.

L. Morales, R. Rodriguez-León, A. S. A. S.-V.-N. S.-G. J. G.-G. F. D. J. N. (2019). Metodología de trabajo orientada al usuario. un ejemplo real en g7 innovation: el proyecto imedea. In *Proceedings of Infors@alud*, pages 155–156.

McMillan, C., Hariri, N., Poshyvanyk, D., Cleland-Huang, J., and Mobasher, B. (2012). Recommending source code for use in rapid software prototypes. In *Proceedings of the 34th International Conference on Software Engineering*, pages 848–858. IEEE Press.

Microsoft Expression Blend (Last Accessed: July, 2019). https://www.microsoft.com/es-es/download/details.aspx?id=10156.

Moqups (Last Accessed: July, 2019). https://moqups.com/.

NDT-Navitgational Development Techniques (Last Accessed: July, 2019). http://iwt2.org/.

Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Ease*, volume 8, pages 68–77.

Schwaber, K. and Beedle, M. (2002). *Agile software development with Scrum*, volume 1. Prentice Hall Upper Saddle River.

V. Cid, D. Soto, C. R. M. C. N. S.-G. A. J. F. D.-J. N. (2019). Concepto plm (patient lifecycle management) en el entorno sanitario. una nueva concepción llevada a la práctica. In *Proceedings of Infors@alud*, pages 155–156.

Velthuis, M. G. P. (2014). *Métodos de investigación en ingeniería del software*. Grupo Editorial RA-MA.

Wireframe Sketcher (Last Accessed: July, 2019). https://wireframesketcher.com/.

Zamzar (Last Accessed: July, 2019). https://www.zamzar.com/es/convert/ppt-to-html5/.

Zeplin (Last Accessed: July, 2019). https://zeplin.io/.