User Survey: Increase of Productivity in Software Development Based on the Use of UML (Unified Modeling Language)

WILLERT.
pioneers in embedded software engineering

During the Rhapsody User Meeting 'RUGS 2004', we carried out a survey among the attendees. The objective of this survey was to obtain data on the return on invest when using UML in combination with the tool Rhapsody®. This document presents the results of the survey.

The result can basically be considered positive. The average of all data implies that an increase in productivity of 100% was achieved before using UML, and an increase of 178.3% was achieved after using UML (after an average of 26 months of use).

On average, a visible increase in productivity was achieved after 4.5 months. There were fluctuations in the introduction phase between 1 to 8 months until a visible increase in productivity was achieved. A major factor for fluctuation is the degree of training and previous knowledge. Users without any knowledge of OOP and UML take very long to get proficient with the technology. Based on 5-7 days training and coaching, the period can be reduced from more than 8 months to below 4 months.

The most common responses to the question “If a development manager wanted to introduce UML based on Rhapsody®, what would you recommend him to take special care of?” were (summarized): ‘Sufficient training and probably a few days of project specific coaching’.

Questions / Projects	1	2	3	4	5	6	Average
How long did it take until productivity was increased, compared to previous methods ?	1	3	6	6	3	8	4,5 month
How long have you been using UML and Rhapsody® ?	24	36	50	15	18	12	25,8 month
How do you estimate your current productivity (x%), compared to previous methods (100%) ?	300	120	200	200	130	120	178,3 %
To what extent do you think do you currently exploit the potential of UML in combination with Rhapsody® ?	50	30	60	60	50	60	51,7 %
How many days training and Coaching you have taken up for the entrance	5	10	2	6	4	4	5,2 days
What is your target platform ?	32 Bit	32 Bit	32 Bit	16 Bit	16 Bit	8 Bit	
How many developers worked with UML at the time of introduction ?	2	4	1	2	3	1	2,2 developers
How many developers work with UML for the time being	8	2	3	4	6	2	4,2 developers

Another common answer was the advice to ‘use UML consistently with code generation down to the implementation stage’. Using UML horizontally across the V-model, e.g. only for design or documentation, does obviously not help achieve a tangible increase in productivity. Investments do not amortize because model and code become inconsistent, and developers will soon return to code level development.

The most common responses to the question “What do you consider to be the main advantages when using UML?” were:

- Graphical representation enhances understandability and communication within the team, especially for colleagues not involved in software development.
- Automatic code generation increases productivity
- Consistency of documentation, model and code (comment: applies only if a code generator is used at the same time)

All Answers: (similar answers were summarized)

What do you consider to be the main advantages when using UML?	Number of answers
Higher level of abstraction, thus improved clarity, graphical representation of structure and behavior results in error-free code	8
Graphical representation in UML simplifies communication within the team and with persons not involved in software development	7
Better and understandable documentation	5
Consistency of documentation, model and code	4
Standardized notation – guarantees portability – future-proof know-how – software understandable even for 'non experts' – independence of RTOS	4
Automatic code generation increases productivity and prevents errors	4
Integration of methods and process – use of UML in all process phases (also analysis) – facilitates structured work	3
MDD (Model Driven Development) with animation and simulation possible	2
Clear and uniform software structure	1
Developers think in terms of models rather than code	1
Structured representation of requirements	1

Based on your experience, what should a development manager take special care of if he wanted to introduce UML based on Rhapsody®?	Number of answers
Select a small or delimitable subproject for the introduction phase. For a pilot project, use code generation by all means. Feedback from code generation is very helpful for learning UML.	5
Sending developers on tool training is a must	4
Start off with only a few workplaces in the beginning. Then allow more and more team members to develop software with UML	4
Intensive OO, UML and Rhapsody® training is indispensable	3
Coaching to support the pilot project	3
Always focus on the objective to “generate complete code for a system“, this is the only way to achieve real benefit	3
Understanding the code remains essential	2
Use UML with code generation only if sufficient target performance is available	3
Installation of a development process	

Frame conditions for the survey and evaluation:

- This is the 2nd version of the evaluation, with unclear answers having been clarified by phone. The survey was carried out among UML users who attended the Rhapsody® User Meeting 2004. (Comment: Attendees of user meetings are mainly successful users. Thus, the evaluation looks a bit more positive than the real average).
- Of all 18 answers, only those with unambiguous information were evaluated.
- Answers from individual users were summarized when referring to a common project.
- Only projects with a minimum runtime of 12 months were considered a basis for getting suitable experience in using UML.
- 6 projects met these requirements and were thus evaluated.