WHAT IS AN END-USER SOFTWARE ENGINEER?

STEVEN CLARKE, MICROSOFT CORPORATION

INTRODUCTION

I work in a multi-disciplinary team at Microsoft that is responsible for designing and building the user experience for users using the Visual Studio .Net suite of products. Visual Studio .Net is a large product suite, comprising a variety of software tools such as code profilers, debuggers, bug tracking tools, testing tools, code editors and language compilers. Multiple programming languages are supported, such as Visual Basic .Net, C# and C++.

Given the large variety of tools and languages that are supported by Visual Studio .Net, we are responsible for designing user experiences for a large variety of different users working in a large variety of different scenarios. For example, on one project we might be designing the user experience for building small web based applications while on another project we might be designing the user experience for a team of developers building a large distributed enterprise application. In both scenarios, the users that participate in the scenarios might differ in their work styles and characteristics just as much as the scenarios differ from each other.

To address the challenge of developing a shared understanding of the users that participate in each scenario we have developed a set of personas that describe the work styles, characteristics and motivations that are common to particular groups of people using our products. The personas help us communicate these characteristics by humanizing them, increasing the empathy that team members have for these fictional users.

There are a couple of things that are of particular interest about these personas that I would like to expand upon:

- We need more than one persona to adequately describe the different work styles, motivations and characteristics that we have observed of people using our products.
- We do not differentiate personas on expertise, experience or educational background.

MULTIPLE PERSONAS

We developed the personas by observing people using our products and noting the work styles, characteristics and motivations of each person. Over a period of approximately 12 months we observed people working in our usability labs and in their own workplaces, working in multiple scenarios. After this time, we were able to identify work styles, characteristics and motivations that were common across many of the observations that we had made. These formed the basis for the three personas that we defined.

We developed three different personas which describe the three sets of work styles, characteristics and motivations that we had observed. These personas are briefly described below:

THE SYSTEMATIC DEVELOPER

- Writes code defensively. Does everything they can to protect their code from unstable and untrustworthy processes running in parallel with their code.
- Develops a deep understanding of a technology before using it.
- Prides themselves on building elegant solutions.

Dagstuhl Seminar Proceedings 07081
End-User Software Engineering
http://drops.dagstuhl.de/opus/volltexte/2007/1080

THE PRAGMATIC DEVELOPER

- Writes code methodically.
- Develops a sufficient understanding of a technology to enable them to use it.
- Prides themselves on building robust applications.

THE OPPORTUNISTIC DEVELOPER

- Writes code in an exploratory fashion.
- Develops a sufficient understanding of a technology to understand how it can solve a business problem.
- Prides themselves on solving business problems.

We have been using these personas for four or five years now and have found them to be an invaluable resource in developing a shared understanding of who the user is when designing user experiences.

DIFFERENTIATE ON WORK STYLES, NOT EXPERTISE

One of the big challenges we've faced in spreading the word about these personas throughout Microsoft (and amongst our own customers) is correcting the assumption that the three personas describe developers with different levels of skill and educational backgrounds. We chose to represent work styles, motivations and characteristics as these are less liable to change over time as opposed to levels of expertise, educational background etc. Our observations have shown us that the work styles we described in the personas are shared by people with varying levels of expertise and educational background. It is not the case that someone starts out as an opportunistic developer then becomes a pragmatic developer after gaining a certain level of experience and expertise.

TRANSFER OF LEARNING

When developing and describing the personas we did not make a distinction between the job roles or titles of the people that we observed. Instead, we simply made observations of people who said that they used our products or other tools to develop software while at work. Many of these people did not describe themselves as software engineers. The variety of job titles that people used included 'Rocket Scientist', 'Surveyor', 'Customer support' as well as 'Software engineer', 'Software developer' etc. In addition we did not observe any relationship between job titles and work styles.

Given this, it is possible that one or more of the personas we developed would apply equally as well in discussions of end user software engineers. Identifying the commonalities between end user software engineers and so called professional software engineers would help enormously in identifying opportunities for transfer of learning between research focused on either community.

For example, Beckwith et al (2005) describe an investigation into the effect of gender on the effectiveness of end user debugging features and report that females were less willing to use new debugging features than males. In addition, females spent their time editing spreadsheet formulas as opposed to learning how to use the new debugging features. These results are similar to observations we make of opportunistic developers who focus on solving the business problem rather than learning how a particular feature works. The challenges are the same for both groups – how to encourage the use of tools that will help solve the business problem.

REFERENCES

Beckwith, L., Burnett, M., Wiedenbeck, S., Cook, C., Sorte, S., and Hastings, M.: Effectiveness of end-user debugging software features: are there gender issues? *ACM Conference on Human Factors in Computing Systems,* April 2005