A Survey of Software Visualization Practices in the Software Development Industry

Craig Anslow, James Noble, and Stuart Marshall
School of Mathematics, Statistics, and Computer Science
Victoria University of Wellington
{craig,kjx,stuart}@mcs.vuw.ac.nz

1 Introduction

The aim of this survey is to provide an overview of the software development industry. The focus of the survey is on the use of software visualisation techniques and tools for software comprehension. The next section lists the questions we will ask in the survey questionaire to the participants.

2 Survey Questionnaire

2.1 User Demographics

1. What is your gender? (male or female)
2. What is your age? (1-18, 19-25, 26-35, 36-45, 46+)
3. What is your occupation or role? (software developer, web developer, programmer, software engineer, database administrator, software tester, user experience, middle management, senior management, other - please specify)
4. How long have you been or worked in a technical role? (0-6 months, 7-12 months, 1-2 years, 2-5 years, 5-10 years, 10+ years)
5. How long have you been working for your current employer? (0-6 months, 7-12 months, 1-2 years, 2-5 years, 5-10 years, 10+ years)
6. What is your highest academic computer related qualification? (None, BIT, BSc / BE, Postgraduate certificate or diploma, Masters, PhD, other - please specify)
7. What professional computer related qualifications do you have? (Java Certified, Microsoft Certified, Network and Systems Administration Certified, other - please specify)
8. What programming languages do you have experience with? (Cobol, Java, C++, C, C#, .Net, Visual Basic, Ruby on Rails, PHP, Perl, Python, SQL, HTML, Java Script other - please specify)

9. What integrated development environments (IDE) do you use? (Eclipse, NetBeans, Microsoft Visual Studio, other - please specify)

2.2 Company Demographics

1. What is the primary business of your company?
   - **Services**: one-offs, customised, mass production
   - **Products**: one-offs, batch jobs, mass produced

2. How many people work for the company in your country? (0-10, 11-50, 51-100, 100+)

3. How many people work for the company globally? (0-10, 11-50, 51-100, 100+)

4. How many, if any, work in software development? (0-10, 11-50, 51-100, 100+)

5. Can you please give a few examples of what (products/services) your company (produces/provides)?

6. Does your company develop software (at least part of) its business? (yes, no)

2.3 Software Development


2. What type of software development projects are undertaken by your company?
   - **Specific software products for customers:**
     - O : One-off contracts
     - M : Mass production (Shrink wrapped)
     - C : Customizing pre-developed software
     - S : Service/support
   - **In house development to support running of organization:**
     - IO : Own development
     - IC : Customization of bought-in products
   - **Product support. Inclusion of software in organisation’s products:**
     - PI : In house development
     - PB : Brought from outside source
3. What kind of software do you develop? (e.g. embedded, client/server, web based, enterprise etc.)

4. What size systems do you develop?

5. How long do your software development projects last for? (0-3 months, 6-12 months, 1-2 years, 2-3 years, 3+ years)

6. Describe the phases a typical development (with you as a developer) project goes through.

7. What proportion of effort is spent, typically, on the different phases of a project? (Requirements, Specification, Design, Implementation, Integration, Testing, Installation, Maintenance then selections of 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 percent)

8. How do you express your requirements?
   (a) No explicit process and no formal language.
   (b) Clear phases, though any method used is implicit and no formal language used.
   (c) Clear phases, and a sequence of informal specifications made during a project.
   (d) Formal process, with semi-formal notation such as UML.
   (e) Formal process, with full-formal notation such as Z and VDM.

9. What software development methodologies do you use? (Waterfall, RUP, RAD, Spiral, Iterative, Cleanroom, Extreme Programming, Scrum, Agile, other - please specify)

10. What tools do you support your development with? (e.g. CASE, database controls, interface generators, source control)


13. How do you record and respond to changing user requirements?

14. Are there standards you have to meet? (Industry standards, ISO 9000 standards, Country Standards, Capability Maturity Model, other - please specify)

15. What quality control mechanisms do you use during your software development processes? (Walkthroughs, Testing Regimes, Code Reviews, Design Metrics, other - please specify)

16. What testing practices do you follow?
(a) Developers perform their own testing during development. Testing by eventual users (before release); includes integration tests on site. Testing by customers (after release); includes beta releases.

(b) Unit, systems and/or acceptance testing. Project-specific test plans used. Regression testing.

(c) Testing done by dedicated testers (not developers involved in the project). Test plans derived from the specification.

(d) Other - please specify.

2.4 Software Comprehension

1. When maintaining a system or using a new piece of software which of the following activities are you most interested in (Likert scale 1-5, with 1 being least and 5 being most)?

   - Investigating the functionality of (a part of) the system
   - Adding to or changing the system’s functionality
   - Investigating the internal structure of an artefact
   - Investigating dependencies between artefacts
   - Investigating runtime interactions in the system
   - Investigating how much an artefact is used
   - Investigating patterns in the system’s execution
   - Assessing the quality of the system’s design
   - Understanding the domain of the system

2. When comprehending software what strategies do you follow (please select all that apply)?

   - Bottom-Up
   - Top-Down
   - Knowledge-Based
   - Inquiry episodes
   - Systematic and as-needed
   - Integrated approaches

3. When trying to comprehend software which of the following questions do you ask (please select all that apply)?

   - What is the class structure of the software system?
   - What interactions occur between objects?
   - What is the high-level structure/architecture of the software system?
• How do the high-level components of the software system interact?
• What patterns of repeated behaviour occur at runtime?
• What is the load on each component of the software at runtime?
• What impact will a change made to the software system have on the rest of the software system?
• What are the data structures that are used in the software system?

4. What specific reverse engineering questions do you ask (please select all that apply)?

• What are the collaborations between the objects involved in an interaction?
• What is the control structure in an interaction?
• How can a problem solution be mapped onto the functionality provided by the software system?
• Where is the functionality required to implement a solution located in the software system?
• What alternative functionalities are available in the software system to implement a solution?
• How does the state of an object change during the interaction?

2.5 Software Visualisation

1. In your opinion which of the following techniques do you classify as software visualisation? (UML diagrams, MS Excel chart types, colour coding in an IDE, pretty printing, call graphs, dependency graphs)

2. Do you use any of the following software visualisation tools? (list the 44 tools from the Software Visualisation text book http://www.eposoft.org/svbook) If not do you use any other tools?

3. If you have used any software visualisation tools what stages of the development lifecycle have you used them in? (Requirements, Design, Implementation, Integration, Testing, Installation, Maintenance - note list the tools with the development categories on the other axis)

4. What kind of software artifacts do you visualise with these tools? (Source Code, Module Dependencies, Object Models, Call Graphs, Run-time Information, Software Archives, other - please specify)

5. What kind of visualisation techniques do you use to visualise these artifacts? (Network / Node-link Diagrams, Line Graphs, Bar Graphs, Pie Charts, Scatterplots, Matrix Charts, Stack Graphs, TreeMaps, TagClouds, 3D, other - please specify)

6. Do the visualisations provide any user interaction? (If so please elaborate)
7. Do the visualisation provide any time-based animation? (If so please elaborate)

8. How important is software visualisation to software maintenance and the measurement of software quality? (Absolutely Necessary, Important but not critical, Relevant, Can do without but is nice to have, Not an issue at all)

9. If you are developing tools for software maintenance, reverse engineering, re-engineering, or software metrics, are your main development interests related to software visualisation? (primary interest / substantial part / every now and then / just using or integrating / not at all)

10. If you are developing tools related to software visualisation, what visualization techniques, toolkits, and layout algorithms are you using? (e.g. specific information visualization metaphors, information visualization toolkit, prefuse, pajek, LGL, Aisee, tree layout, force-directed, other - please specify)

11. What do you consider the specific needs and challenges for software visualisation in the context of software development and software maintenance?

12. What kind of software visualisation tools and techniques would you like to have?

13. Where would you like to have software visualisations displayed? (inside IDE, separate desktop application, web application, other - please specify)

14. What percentage do you think other software developers would use software visualisation tools and techniques on a regular basis? (10, 20, 30, 40, 50, 60, 70, 80, 90, 100 percent)

15. What percentage do you think other software developers have used software visualisation tools and techniques? (10, 20, 30, 40, 50, 60, 70, 80, 90, 100 percent)

2.6 Contact Details

If you would like to go into the draw for winning our survey prize of XYZ, please supply your contact details. Please also indicate if you would be available for a possible follow up interview.

1. Name:
2. Email:
3. Phone:
4. Interview: (Yes / No)

Acknowledgments

This work is supported by the New Zealand Foundation for Research Science and Technology supported Software Process and Product Improvement (SPPI) project, and a TelstraClear post-graduate scholarship.