"Does it work?" The Art and Science of System Resilience

Daniel Axtens
u5376292@anu.edu.au

May 10, 2013

Abstract

System resilience is about ensuring that the system produced matches the system requirements. It focuses on various forms of testing. A key motivator is cost—the earlier a problem is identified, the cheaper it is to fix.

The experience of a team at Carnegie Mellon University in developing wearable computers is considered. Their experience shows that (at least in the context of highly experimental products pushing the bounds of what is possible) rapid prototyping **works**.

At this stage in our project (Silent Alarm), only analysis and Type I testing are possible. Analytical, formal validation is outside our collective skill-set but rapid prototyping would be practical and valuable. If we were able to do formal verification, it would be best applied iteratively with the prototyping.

The Type I–IV progression is likened to the Waterfall model for software development: this is compared to the spiral model for software development, and a more iterative model of testing is advanced and the implications discussed.

1 Background

System resilience is about ensuring that the system requirements match the system actually produced. As such, it focuses on various forms of testing. This set is drawn from Blanchard & Fabrycky (2011).

Analytical CAD/CAM, formal verification of specification. May include analytical devices such as Petri nets and statistical models (Billinton & Allan 1983).

Type I/Proof-of-concept *"Do the bits work?"* Testing individual parts of the system, often in very early stages of development—for example breadboards. Rapid prototyping sits within this stage.

Type II/Model *"Does the prototype work?"* Testing the system as a whole but in isolation from the environment. It is usually full prototypes or early production models that are tested.

Type III/Operational *"Does it work in the real world?"* Testing the full system in situation as close to real-world usage as possible.

Type IV/Support *"Can we improve it?"* Testing after deployment, usually to identify areas for improvement either in the system itself or in its usage and maintenance procedures.

A key motivation is cost: it is cheaper to fix a problem in the proof-of-concept stage than in operational testing (Billinton & Allan 1983).

This sort of development is especially suited to software development, leading to slogans such as "Release early, release often. And listen to your customers." (Raymond 1999), the emergence of Test Driven Development and Behaviour Driven Development methodologies, and the development of some amazingly powerful software testing frameworks.

2 Literature Review

Smailagic, Siewiorek, Martin & Stivoric (1998) is a case study of rapid prototyping (part of Type I testing). Two decades before Google Glass, a team at Carnegie Mellon University was developing wearable computers. Their experience shows that (at least in the context of highly experimental products pushing the bounds of what is possible) pervasive rapid prototyping was effective at producing better products for less.

Two 'development cycles' are compared, where they developed different models of wearable computers. In one cycle, they built the Navigator 1, which was put together from COTS (Commercial Off-The-Shelf) components. In the second cycle they built the VuMan 3, which was custom-designed with rapid prototyping. Table 1 shows the differences.

Attribute	Navigator 1 (COTS)	VuMan 3 (custom)
Overhead factor (%)	56.5	5.6
Cost ($)	4840	3550
Person power (months)	28	23
Software portability	95%	35%
Power (W)	7.5	1.5

Table 1: Attribute comparision between Naviagator 1 and VuMan 3. "Overhead" represents the percentage of discrete functions included in the components making up the system which are not used in the system.

This shows that the further you push rapid prototyping, the better results: the custom model was better, cheaper, and produced faster. In particular, there was a ten-fold reduction in overhead.

This further supports the assertion that rapid prototyping is particularly useful for novel, experimental and untested concepts. For example, it's much more applicable to building a phone or a wearable computer or a piece of software to building a road or a bridge or a tunnel.

Given that we are developing an integrated hardware/software solution, and that we are developing something that, as far as we've been able to tell, is not a well-explored problem, rapid prototyping is a valuable tactic.

3 Application to Project

At this stage of our project (Silent Alarm), only analysis and Type I testing are available to us.

Analysis A number of analytical, formal validation tools are applicable. This is outside our collective skill-set, but as we're designing a safety device, it would be appropriate to do so if we were building this as a company.

For example, we want to analytically prove that:

- The software will:
  - Never 'crash' or 'hang'.
  - Always respond within an acceptably short time period (hard real time).
- The radio protocol is robust against:
  - expected levels of interference.
  - multiple systems and units operating within range of each other. If a neighbouring warehouse installs the same system, we don't want cross-talk.

This is analytic testing because it is not done by actually testing systems. For example, the radio protocol is tesed by someone with mathematical expertise proving mathematically that the protocol satisfies certain properties. Unlike Type I testing, analytical testing of these properties *proves* that the properties will *always* be satisfied.

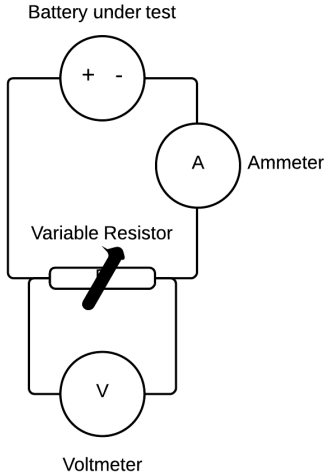
Type I More practically, we could apply rapid prototyping (in the Type I tier). There are a number of aspects well suited to 'proof of concept' prototyping. For example, we could test the range of the radio modules and the geometry, thermal and electrical characteristics of a poser supply.

For example, testing various battery chemistries and configurations is best suited to Type I testing. Simulation can explore electrical properties, but simulating thermal properties and getting a feel for weight and shape is much harder.

Type I tests are often straightforward. Here, we need two multimeters (one to measure voltage, one to measure current), themometer, the battery, and a variable resistor, as shown in Figure 1.[1]

This test would be done by someone with basic understanding of electronics.

If the battery can consistently deliver the required current for the vibration motor, never drops below the microcontroller core voltage and doesn't get too hot, it passes.



Sample Method:

- Set the resistor to deliver standby current. Record voltage.
- Wait 30 minutes.
- Record voltage and temperature.
- Reduce resistance to deliver motor equivalent current for 30 seconds.
- Record voltage and temperature, reset resistance to standby level.
- Repeat until the battery is flat or 8 hours have passed.

Figure 1: Test setup and procedure for battery discharge characteristics.

[1] This is not a perfect model (a better model would include an inductor and a resistor. However, given the DC nature of the circuit and the infrequent utilization of the motor, this should be sufficient.

Influence of tests Discoveries at this stage are particularly influential on the design. If the size and shape of our wearable module conflicts with the antenna geometry of our radios, we may need to change radio technology. Similarly, the weight restriction could force a rethink of power supply designs. Alternatively, we may need to rethink how and where our module was worn.

Furthermore, some things we develop for prototyping—such as an easily removable power source—may be integrated as-is in the final design.

Interaction between Analysis and Type I Formal verification would be best applied iteratively with prototyping. Formal verification does not prove usefulness—to borrow an analogy from McConnell (1993), it will tell us that we're hitting a target, but not that we're hitting the right target. Prototyping won't prove that we're hitting the target every time and in all circumstances, but helps us aim for the right one.

4 Discussion

There is an overlap in thinking between the model of Blanchard & Fabrycky (2011) and the Waterfall model of software development, as shown on the left of Figure 2 (McCormack & Conway 2005).

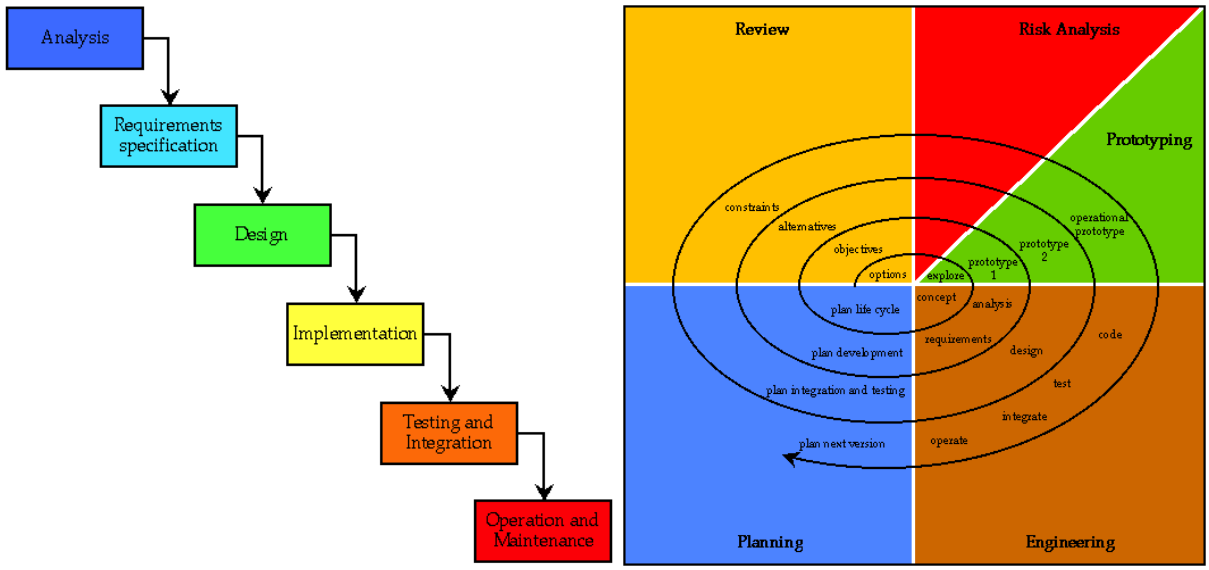


Figure 2: The Waterfall Model (left) and the Spiral Model (right)

In the waterfall model, stages flow from one to another without any feedback. Similarly, the Type I–V model also supposes a linear flow from one type of development (and therefore testing) to another, with little expectation of feedback.

The waterfall model is a poor fit for the software lifecycle (McConnell 1993, McCormack & Conway 2005). Its weaknesses are probably also applicable to engineering problems where requirements are unclear and technology is unproven.

Consequence, alternative models such as the spiral model (Figure 2, right) have been developed. Application of this to a 'conventional' engineering project would mean that a project cycles through analysis and Types I through III several times.

This requires a change in thinking about testing. Testing is often seen in terms of fault-finding or avoiding the expense of late changes (Billinton & Allan 1983). The design comes first, testing comes afterwards.

Adopting a spiral model requires a change to seeing testing as a core part of design. If 'doing is the best kind of thinking' (Chi 2013), testing must shape our design, not just in Type I, but throughout the process. We must *always* be willing to let a practical insight take us back to the drawing board. This helps keep the focus on meeting customer requirements, the core of system resilience.

5 Conclusion

System resilience is ensuring that the system meets customer requirements. It centres around testing. Blanchard & Fabrycky (2011) identifies 4 types of testing: proof-of-concept, prototype, operational and maintenance. Analytical/formal validation may precede type I where applicable.

Rapid prototyping (part of type I) reduces cost and improves quality. The more that rapid prototyping is allowed to pervade design, the better the results.

Type I testing is directly applicable to our project. Analytical validation would also be valuable, especially applied iteratively with Type I.

The linearity of the testing model was queried. Adoption of a spiral model helps keep the focus on meeting customer requirements, not just finding defects early.

References

Billinton, R. & Allan, R. N. (1983), *Reliability evaluation of engineering systems*, Plenum Press, New York.

Blanchard, B. & Fabrycky, W. (2011), *Systems Engineering and Analysis*, 5th edn, Pearson, New Jersey.

Chi, T. (2013), 'Rapid prototyping Google Glass'.

McConnell, S. (1993), *Code complete: a practical handbook of software construction*, Microsoft Press, Redmond, WA.

McCormack, J. & Conway, D. (2005), 'CSE2305 Topic 13: The Software Development Process', `http://www.csse.monash.edu.au/~jonmc/CSE2305/Topics/07.13.SWEng1/html/text.html`.

Raymond, E. S. (1999), *The Cathedral and the Bazaar*, 1st edn, O'Reilly & Associates, Inc., Sebastopol, CA, USA.

Smailagic, A., Siewiorek, D. P., Martin, R. & Stivoric, J. (1998), 'Very Rapid Prototyping of Wearable Computers: A case study of VuMan 3 Custom versus Off-the-Shelf Design Methodologies', *Design Automation for Embedded Systems* **3**(2-3), 219–232.
URL: *http://dx.doi.org/10.1023/A%3A1008850609458*

6 Peer Review

6.1 One

Demonstrates that formatting requirements have been met:

Only two things drew my attention about the format according to

http://users.cecs.anu.edu.au/ u3951377/ENGN2225/lib/exe/fetch.php/assessment/formattingyourreport.pdf

- the figure should be referenced together with the figure caption

- 1.5 line spacing

Demonstrates a correct understanding of the theory:

The references are strong and the content is very solid, the understanding is clearly presented in the research paper. My only critisism will be on an iten of Table 1. "Overhead factor". It is not clear what exactly is and how it contributes for the argument.

Application of the theory to the project:

I have only a technical critic on the application. The testing conditions and simulations should be as close as possible to the real model, the resistive loan ignores the real model of a motor (a resistor with a inductor both in series)

Quality and relevance of bibliography:

The content of the reserch paper is rich and the bibliography is unquestionable comprehensive, but I think there are too much references for only 5 pages.

Suggestions on how could the paper be improved:

This research paper is very hard to peer review in terms of finding wrong things. I guess all the improvements that I could give were mentioned on the comments above.

Andre Dzis Giacomini

6.2 Two

Demonstrates that formatting requirements have been met:

1. Page must be A4. YES
2. Headings and sub-Headings are in a front size larger than the text body. YES
3. 2cms margins. YES for the text but NO for figure 2, figure 2 exceeds the right margin
4. 1.5 line spacing. YES
5. 12-point front size. YES
6. Test justified. YES
7. Australian English and metric units. YES
8. Figure referred to in the text. YES
9. Figure left aligned on page. NO, figure 2 right aligned on page

10. Figure caption with description. YES
11. Figure reference. YES, figure 1 is your own figure, therefore, no reference is required in that case
12. Footnote used. No apparent need to
13. Page number. YES

Demonstrates a correct understanding of the theory:

The author demostrates a correct understanding of the theory, as an extensive background that explains different testing process at different levels. In addition, a case study has been provided which shows the performance outcome of the product with rapid prototype is better than the product without rapid prototype. However, by looking at the application of the theory part of the paper, the author kind of missed some of the important criteria of system resilience. For instance, the author didn't mention any thing about the pass/fail criteria. The author mentioned about plot. But what kind of plot implies a pass or fail???

Application of the theory to the project:

I would give a 6/10 for this part. Analytical testing is testing of the design by using software like CAD or test by having a software simulation. Actually, the author can use PSPICE program to obtain the current/voltage plot for analytical part. The author kind of messed up between analytical and type 1 analysis. Robustness against multiple existing system within range should fall under type 1 analysis instead of analytical part. The author describes the test procedure in the paper, which is good. However, the author didn't mention about the test person and pass/fail criteria.

Quality and relevance of bibliography: Most of the references are of high quality and highly relevant, for instance:

1. Blanchard & Fabrycky (2011) mentioned SR is about ensuring the system requirements match the system actually produced. YES, that is the whole point of SR.
2. BIllinton & Allan (1983) mentioned statistical model used in analytical testing. YES, analytical carry out rapid prototype.
3. However, Tuan, Zheng & Tho (2010), traditionally, pacemaker used software testing techniques to find bugs in implementation of system. YES, That relate to the analytical.

Suggestions on how could the paper be improved:

1. Explain everything in the text, don't just leave that without explanation. For example, the author referenced Tuan, Zheng & Tho (2010) provides an interesting discussion of pacemaker, but what are the interesting discussion?
2. The author used a lot of abbreviation but failed to mention the actual meaning. For example, what does CMU means in the text? I suggest the author at least write down the full text once if trying to make the paper shorter.
3. Overall very good but review section 3, application of project. I suggest the author to mention all the testing processes in detail, go in depth for one particular attribute but not roughly describe a few attributes.