

Unsolved Problems in Requirements Engineering

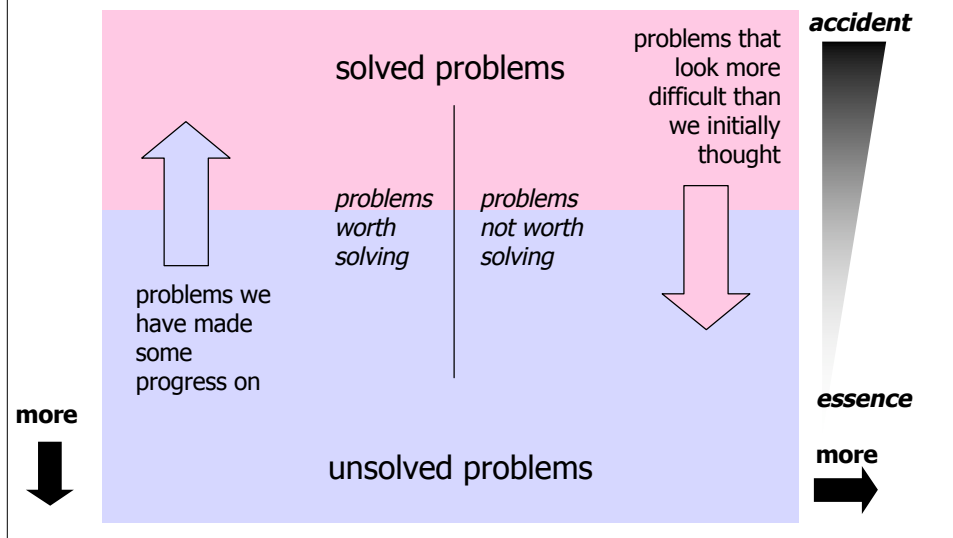
Anthony Finkelstein
University College London
Department of Computer Science
and
London Software Systems



Warning! Warning! Warning!

This is a personal view

Problems, Problems ...



Who ...

- 'We' may not have solved the problem
- It may not be 'our responsibility' to handle the unsolved problems
- Despite the fact that my presentation is largely a catalogue of unsolved problems I am bullish about our prospects and achievements
- Rough code
 - Solved
 - Partially Solved
 - Unsolved

Grand Challenges

- What they are and the benefits they can bring
- Why they are not necessarily the 'be all and end all' of a future research programme
- Where we can hope to take advantage of them



Solved [?]

- We have the component parts of sound industrial strength requirements engineering processes appropriate to varying scales of development
 - Modelling techniques
 - Tools
 - Practices and standards
- We are less sure how to assemble these components into a coherent development process
- We have not been able to persuade enough practitioners of the benefits of adopting them

Solved [?]

- This last may be attributable to two unsolved problems:
 - We don't have a clear 'economic theory' of requirements engineering that allows us to reason about it's costs and benefits [see later]
 - We don't have proven educational and training methods and materials to support the transfer of knowledge in this area



Solved [?]

- We have made significant progress on the application of formal modeling. We can build significant sized formal specifications and we can analyse them
- Theorem proving still requires very high skill levels
- We don't have a good idea about how to formulate properties to reason against

Solved [?]

- We have developed (or judiciously borrowed) a range of elicitation techniques
- These techniques are largely research tools that have not been widely applied, used in combination or at scale

Solved [?]

- We have made very significant progress on expressing and reasoning about non-functional properties such as reliability, security, performance
- There is a lot further to go ...
- And we run slap bang into a really BIG unsolved problem
 - Compositional reasoning! Which given that virtually all the systems we want to build are subject to change and extension suggests a major headache

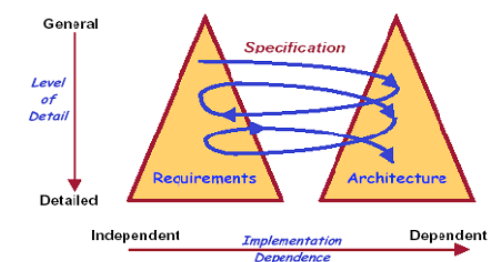


Unsolved [?]

- Lip service is paid to the notion of requirements engineering as a lifecycle wide activity yet most of the work that has been done is based on implicit 'up-frontness'
- We need to build requirements engineering processes that are better integrated with, more accurately stitched into the fabric of the wider software development process

Unsolved [?]

- We still do not understand the relationship between requirements and system architecture
- We know however that it is the cornerstone of software development
- We need to make 'twin peaks' more than a drawing



Unsolved [?]

- We don't have a good understanding of how to 'bound' systems
- So we make ad hoc decisions resulting in either setting the scope of investigation too narrowly and over constraining the solution or extending the scope too widely and consuming excessive cost on gathering incomplete requirements



Unsolved [?]

- We are still very poor at resource estimation. We are unable to reliably predict the cost/effort required to build a system.
- We may be fortunate and have built a very similar system before. Function points are precious little assistance.



Unsolved [?]

- If, as not infrequently happens, budgets get squeezed, deadlines shortened or there are time overruns it may be necessary to scale-back requirements engineering processes. We do not know how to do this
- More particularly we do not know very much about how this affects project risk




Unsolved [?]

- Requirements are rarely cut and dried. Rather there is an assemblage of competing priorities, preferences and expectations incompletely marshalled by a complex set of stakeholders fulfilling different roles
- We have fiddled round the edges of this problem borrowing stuff haphazardly from decision theory. We need to find new ways of handling this that makes the notion of value a first class object within the requirements engineering process

Unsolved [?]

- We know that change is a fundamental feature of all software development processes and yet our understanding of requirements volatility, both its nature and consequences, is negligible
- We must achieve an understanding sufficient to be able to develop accurate impact analysis procedures and tools



<..(change)..>

Unsolved [?]

- There are some 'new' classes of systems that we have a particularly poor understanding of how to establish and manage the requirements for
- These include but are not restricted to
 - Systems that must adapt to context
 - Systems embedding significant COTS components
 - Systems that involve user scripting and 'pluggability'

Unsolved [?]

- We would like to build systems that exhibit requirements reflection. In other words systems that are able dynamically to give an account of their requirements and the extent to which they have satisfied them
- Run-time monitoring is a contribution to this though it is only a small part of the story



Unsolved [?]

- We need to preserve knowledge and experience gained in requirements engineering so that it can be used in similar situations
- There have been numerous attacks on this problem - domain modelling, requirements cliches, componentised modelling schemes all partial and all more or less flawed
- The most convincing is problem frames but this is merely a sketch of a possible approach

Unsolved [?]

- It is well known that individuals exhibit large differences in programming ability - only partly correlated with training and experience. Anecdotally the same applies to requirements engineering (and software architecture). It would be good to understand this better ...
- Might help us to understand the activity better and get a cheap productivity boost

Unsolved [?]

- The stakeholder 'stuff', finding, meeting, interpreting

Conclusion

- There is much to do!
- We need to have the courage to tackle big questions and to step out of existing conceptual frames
- We have been good at borrowing (and this is an effective strategy) but the problems we have to tackle now may not yield to it