



Managing the operational risks of user-developed software

Louise Pryor
www.louisepryor.com



Outline

- What are we talking about?
- What are the risks?
- Managing risks
- Development process and standards
- The significance of bugs

See the paper

www.louisepryor.com



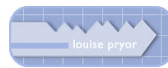
Main point: 1

- The problem with bugs is not just that they may exist, but also the effort that goes into
 - finding
 - fixing
 - and preventing them



www.louisepryor.com

Caution: User-developed systems may be hazardous to your organization Davis 1982



Main point: 2

- The most effective risk management for user-developed software is done at the design and development stage
 - It's worth spending time at the beginning...

www.louisepryor.com

The largest portion of large-scale end user applications today involve spread-sheet development Panko, 2000



User-developed software

- Spreadsheets
- Personal databases
- VB programs
- Parameter-driven modelling systems
- DOS scripts
- ...
- Combinations of all of the above



www.louisepryor.com

End users are putting their companies at risk by setting up spreadsheets...



Uses of user-developed software

- Pricing
- Reserving
- Financial reporting
- Internal planning
- Modelling
- ...

www.louisepryor.com

... without realising that this demands the discipline of traditional programming. Kavanagh 1997

Example

A reserving system

- Reserving package reads text files, writes to files
- Spreadsheets for final analysis
- Spreadsheets consolidate results for all lines

www.louisepryor.com *Our findings are disturbing...*

Risk management process

- Identification
- Assessment
- Mitigation and control

www.louisepryor.com *...as 78% of models (i.e. spreadsheets) had no formal quality assurance...*

Identification

- Specify the behaviour that is expected
- Look for possible deviations, their causes and consequences
- Specification of requirements should be done at the design stage anyway
 - Correctness
 - Usability
 - Availability
 - Maintainability
 - Performance

www.louisepryor.com *... to ensure they were built to specified requirements and were fit for the purpose. Kavanagh 1997*

Example requirements

- Turnaround time
 - whole reserving process
 - individual stages
 - maintenance and enhancements
- Formulae and algorithms
- Data flow
- Who is expected to be able to perform each function

www.louisepryor.com A KPMG survey found

Maintaining user-developed software

- A major use
 - Often a reason for using it in the first place
- Requirements should be specified in the same way
 - Turnaround times
 - Usability
 - ...

www.louisepryor.com *95% of models were found to contain major errors (errors that could affect decisions based on the results of the model)*

Example risks

Cause	Wrong results	No or late results
Bugs in formulae	✓	✓
Bad data flow Import wrong text file Link to wrong part of other spreadsheet	✓	✓
User error in input	✓	✓
Calculation takes too long		✓
Modification takes too long		✓

www.louisepryor.com *59% of models were judged to have 'poor' model design*



Causes and consequences

- Need to step back through causal chain
- What can cause bug in formula?
 - Typographic error
 - Cutting and pasting
 - Change not made to all occurrences
 - Misuse of absolute/relative
 - Faulty assumption (eg data sorting in lookup)
 - Misunderstanding of built in function
 - ...

www.louisepryor.com

92% of those that dealt with tax issues had significant tax errors

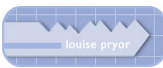


Mitigation and control

- Prevention
 - Look at the causes from the identification stage
 - Stop cause from happening
 - Sever link between cause and consequence
- Deal with problems when they have happened
 - Look at consequences from the identification stage
 - Recognise occurrence of consequence
 - Take remedial action
- Ignore

www.louisepryor.com

75% had significant accounting errors Chadwick 2000

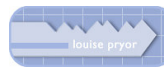


Effective control

- Easier to stop problems happening than to fix them later
- Some can't be fixed
 - If the process has already taken too long...
- Must budget for problem detection
 - How can you tell if you've got the wrong results?
 - Have to look for problems even if you think you've prevented them

www.louisepryor.com

The use of spreadsheets in business is a little like Christmas for children.



Example: prevention

- Techniques to avoid bugs in formulae include
 - Don't use overlong or complex formulae
 - Don't duplicate calculations
 - Use array formulae to prevent mis-copying
 - Document assumptions (using cell comments)
- Early detection is vital
 - Spot errors before they do any harm
 - While you still know what changes were made

www.louisepryor.com

They are too excited to get on with the game to read or think about the 'rules' which are generally boring. Finch 2001



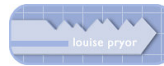
Example: Remedial action

- The problem is spotting the problem
- Requires good testing techniques
- Should be performed during development anyway



www.louisepryor.com

Even in a domain such as indirect taxation, which is characterised by relatively simple calculations,



Building good software

- Specify
- Document
- Test
- Review



www.louisepryor.com

relatively high domain knowledge by developers, and generally well-documented calculation rules,



Example of process

- Specify purpose of modification
- Approve modification
- Take copy of current (working) version
- Make changes to copy
- Test changed version
- Review changes
- Replace current version with changed version

www.louisepryor.com

the use of spreadsheet applications is fraught with danger and errors.
Butler, HM Customs & Excise, 2001



Testing and review

- Testing
 - Does the system do what it is meant to do?
 - Shows existence of problems
- Review
 - Will the system do what it is meant to do?
 - Does the system adhere to standards?
 - Indicates how to fix potential problems

www.louisepryor.com

Spreadsheet errors are a business time-bomb waiting to go off.
Anon, cited by Chadwick 2001



Throughout ...

- Keep changes small
 - Test them individually
 - Easier to reverse out of them
- Have standards for coding and documentation
 - Saves having to think all the time
 - Makes reviewing easier

www.louisepryor.com



... and always

- Think about the risk issues
 - What could go wrong?
- Design spreadsheets with usability and maintainability in mind
- Make assumptions explicit
- Include checks in the code
- Test early and test often

www.louisepryor.com



Conclusions

- Better to avoid bugs and other risks than to take remedial action
- Take action at the development stage

www.louisepryor.com