National Physical Laboratory

Do tools influence the choice of programming language?

Graeme Parkin

graeme.parkin@npl.co.uk

Mathematics and Scientific Computing Group

Division of Enabling Metrology

Aim

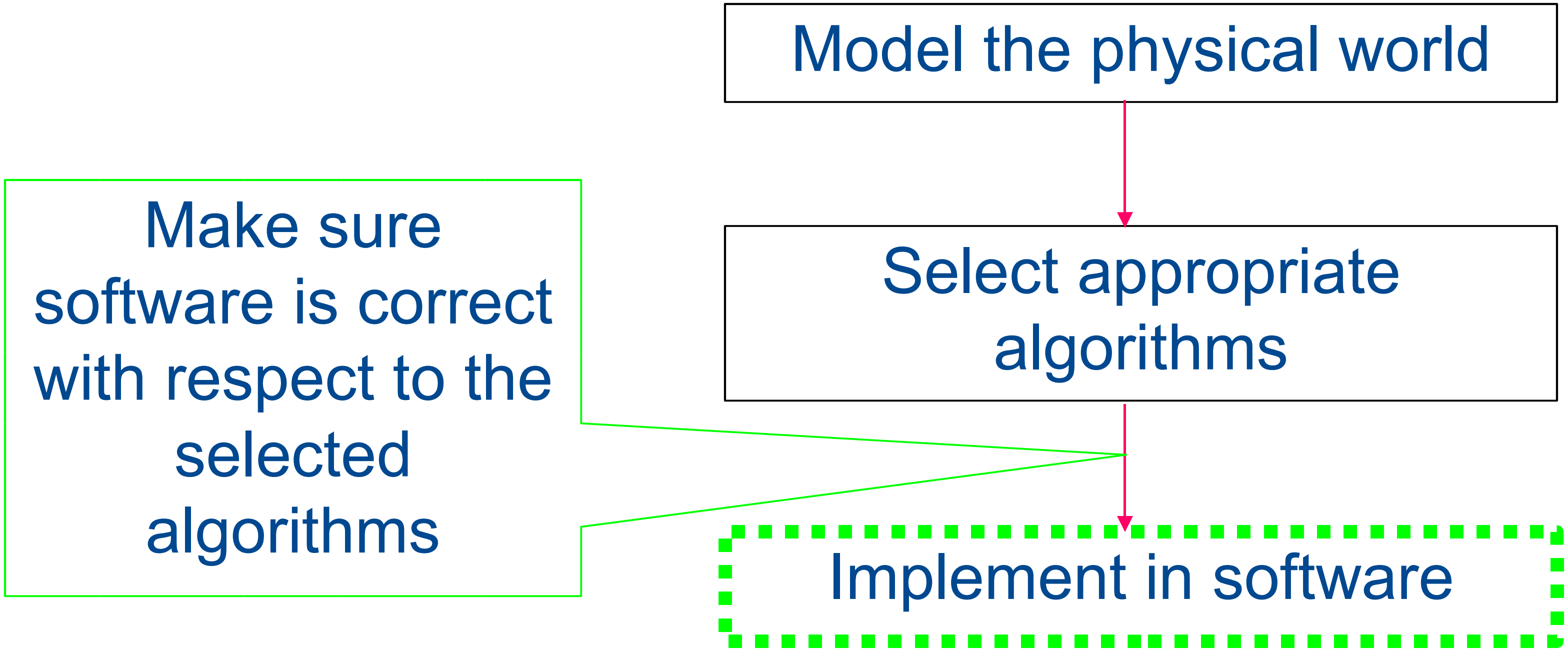
To show:

- That the availability software tools will influence the choice of programming language
- How to determine what software tools are required when developing a particular piece of software in a programming language

Contents

- Scientific software
- Failure of software
- Software problem
- Software fit for purpose
- Example
- Summary

Scientific software



Model the physical world

Select appropriate algorithms

Implement in software

Make sure software is correct with respect to the selected algorithms

Failure of software

- Ariane 5 - *The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value.*
- Saturn's moon Titan - *But a single line of software code omitted from the craft's onboard communications system resulted in the failure of one of its two channels of communication, meaning data on only one channel was radioed back to Cassini.*

Photo:CNN (http://www.cnn.com/WORLD/9606/04/rocket.explode/)

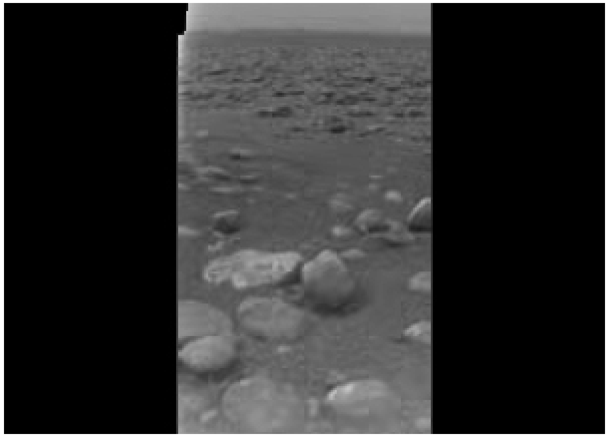


Photo: ESA/NASA/JPL/University of Arizona Image: First images from Titan

Software problem

- Producing software fit for purpose
  - Problem: software cannot in general be shown to be error free, the more effort put in to remove errors the more it costs to develop
  - Solution: use development techniques that produce software that is suitable for where it is to be used
- Software validation is the process of ensuring that the appropriate techniques have been used correctly to produce software fit for purpose

Software fit for purpose

Three steps in our approach:

- A risk assessment, the purpose of which is to make an objective assessment of the likely risks associated with a software error. The assessment considers the risk factors of: criticality of usage, legal requirements, the impact of complexity of control, and the complexity of data processing.
- Assigning a Measurement Software Level (MSL) indicated by the results of the above risk assessment.
- Applying software validation techniques indicated by the assigned MSL.

Software techniques require tools

Most software techniques require tools to support them some examples are:

- IDE (Integrated Development Environment)
- Debuggers.
- Code review checkers (range from making sure naming conventions are applied to checking variables are being used)
- Test harness
- Code coverage
- Static analysis

Further information

The whole process is contained in (freely available):

- Software Support for Metrology, Best Practice Guide No. 1, Validation of Software in Measurement Systems, Brian Wichmann, Graeme Parkin and Robin Barker, March 2004, Version 2.1 http://www.npl.co.uk/ssfm/download/documents/ssfmbpg

Example

Functionality of software to be developed consisted of:

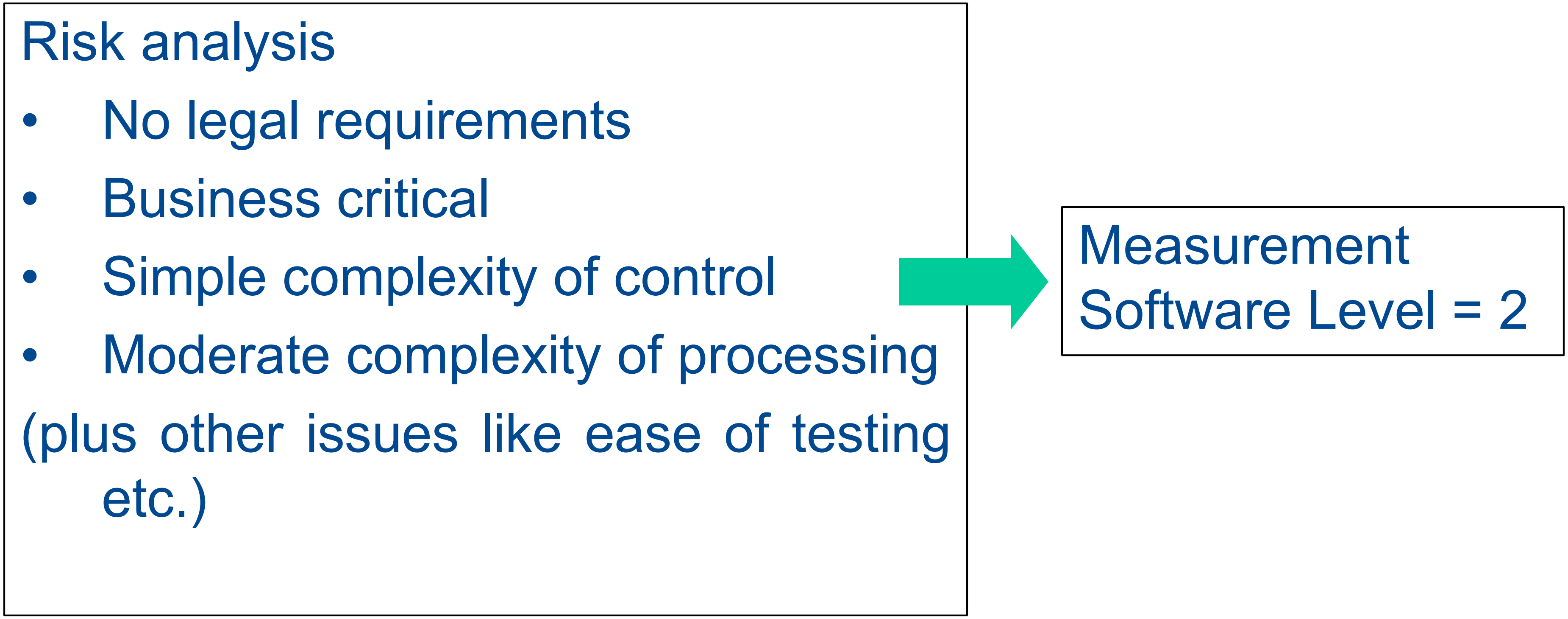
- Apply a Gaussian filter with cut-off wavelength λc to data representing a primary (*P*) profile to obtain data representing waviness (*W*) and roughness (*R*) profiles.
- Evaluate surface texture roughness parameters *Ra*, *Rq*, *Rsk*, *Rku*, *Rp*, *Rv*, *Rz*, *RSm*, *Rc*, *Rt* and their waviness (*W*) and primary (*P*) profile equivalents.
- Read in SMD format

Example - context

Context:

- Reference software
- Platform independent
- Easily accessible

Example – risk analysis



Risk analysis

- No legal requirements
- Business critical
- Simple complexity of control
- Moderate complexity of processing

(plus other issues like ease of testing etc.)

Measurement Software Level = 2

Example – selection of techniques and tools

Ref	Technique	Tools (Java mainly)
12.2	Review of informal specification	–
12.4	Mathematical specification	MATLAB 7.0
12.6	Static analysis	Java compiler 1.4.2_4 Checkstyle 3.3
12.7	Defensive programming	–
12.8	Code review	Checkstyle 3.3
12.13	Statement testing	Clover 1.3_02
12.16	System level testing	–

Example – other tools

Purpose	Tools for Java
IDE to support development	BlueJ 2.0.3
Component testing	JUnit 3.8.1
Java-based build tool - it is kind of like Make.	Ant 1.6.2

Summary

Shown:

- Software tools should be considered when selecting a programming language
- How to use risk analysis in determining the software tools required (see Best Practice Guide No. 1)