Basic concepts of software engineering and maintenance

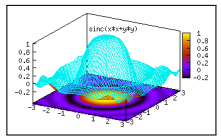
G. Giuliani International Centre for Theorethical Physics - Trieste Earth System Physics Section

ICTP - Earth System Physics Section

Advanced School on Scientific Software Development
Trieste, 20 Feb - 2 Mar 2012

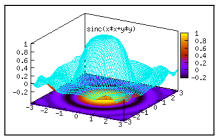
Is Software part of Science?

- What Science is About



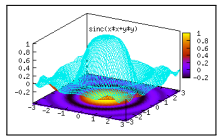
Is Software part of Science?

- What Science is About
  - Try to understand Reality



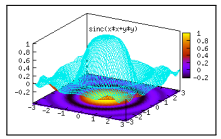
Is Software part of Science?

- What Science is About
  - Try to understand Reality
  - Collect information as numerical values



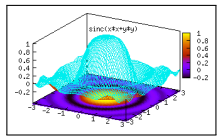
Is Software part of Science?

- What Science is About
  - Try to understand Reality
  - Collect information as numerical values
  - Build knowledge on it with analysis and models



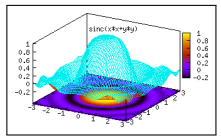
Is Software part of Science?

- What Science is About
  - Try to understand Reality
  - Collect information as numerical values
  - Build knowledge on it with analysis and models
  - Or mix the previous two points



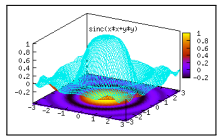
Is Software part of Science?

- What Science is About
  - Try to understand Reality
  - Collect information as numerical values
  - Build knowledge on it with analysis and models
  - Or mix the previous two points
- Computer aided Science



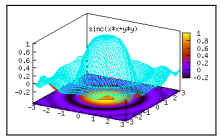
Is Software part of Science?

- What Science is About
  - Try to understand Reality
  - Collect information as numerical values
  - Build knowledge on it with analysis and models
  - Or mix the previous two points
- Computer aided Science
  - Science IS information processing



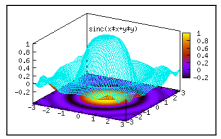
Is Software part of Science?

- What Science is About
  - Try to understand Reality
  - Collect information as numerical values
  - Build knowledge on it with analysis and models
  - Or mix the previous two points
- Computer aided Science
  - Science IS information processing
  - Large computation and data require a computer



Is Software part of Science?

- What Science is About
  - Try to understand Reality
  - Collect information as numerical values
  - Build knowledge on it with analysis and models
  - Or mix the previous two points
- Computer aided Science
  - Science IS information processing
  - Large computation and data require a computer
  - Software development is not science for itself



The Scientifical Problem

- The Scientist aims to undertand the reality

The Scientifical Problem

- The Scientist aims to undertand the reality
- The Scientist makes models of the reality

The Scientifical Problem

- The Scientist aims to undertand the reality
- The Scientist makes models of the reality
  - The model is a bunch of math equations

The Scientifical Problem

- The Scientist aims to undertand the reality
- The Scientist makes models of the reality
  - The model is a bunch of math equations
  - The calculator helps get the math done sooner

The Scientifical Problem

- The Scientist aims to undertand the reality
- The Scientist makes models of the reality
  - The model is a bunch of math equations
  - The calculator helps get the math done sooner
  - The software is the instruction to get there

The Scientifical Problem

- The Scientist aims to undertand the reality
- The Scientist makes models of the reality
  - The model is a bunch of math equations
  - The calculator helps get the math done sooner
  - The software is the instruction to get there
  - If the time to result is lowered by calculator

The Scientifical Problem

- The Scientist aims to undertand the reality
- The Scientist makes models of the reality
  - The model is a bunch of math equations
  - The calculator helps get the math done sooner
  - The software is the instruction to get there
  - If the time to result is lowered by calculator
  - The Scientist is happy with publications

The Software

- Software for science is...

The Software

- Software for science is...
  - Data Digitalization and Storage

The Software

- Software for science is...
  - Data Digitalization and Storage
  - Numerical Algorithms and Packages

The Software

- Software for science is...
  - Data Digitalization and Storage
  - Numerical Algorithms and Packages
  - Programming Languages and Environment

The Software

- Software for science is...
  - Data Digitalization and Storage
  - Numerical Algorithms and Packages
  - Programming Languages and Environment
  - Data Visualization Tools

The Software

- Software for science is...
  - Data Digitalization and Storage
  - Numerical Algorithms and Packages
  - Programming Languages and Environment
  - Data Visualization Tools
  - Word Processors and Publishing

The Software

- Software for science is...
  - Data Digitalization and Storage
  - Numerical Algorithms and Packages
  - Programming Languages and Environment
  - Data Visualization Tools
  - Word Processors and Publishing
  - Information Exchange Systems

My Problem needs a new solution

- I need to write software if

My Problem needs a new solution

- I need to write software if
  - I have a new instrument

My Problem needs a new solution

- I need to write software if
  - I have a new instrument
  - I have a new model to implement

My Problem needs a new solution

- I need to write software if
  - I have a new instrument
  - I have a new model to implement
  - I have a new numerical algorithm

My Problem needs a new solution

- I need to write software if
  - I have a new instrument
  - I have a new model to implement
  - I have a new numerical algorithm
  - I have a new viewing point to my data

My Problem needs a new solution

- I need to write software if
  - I have a new instrument
  - I have a new model to implement
  - I have a new numerical algorithm
  - I have a new viewing point to my data
  - I have a new community to share data with

My Problem needs a new solution

- I need to write software if
  - I have a new instrument
  - I have a new model to implement
  - I have a new numerical algorithm
  - I have a new viewing point to my data
  - I have a new community to share data with
- Is there a Good Way to write Scientific Software ?

The wrong way of writing scientific software

- I will just have to do it once

The wrong way of writing scientific software

- I will just have to do it once
- Nobody cares on how I do get to the answer, I will never share this with others

The wrong way of writing scientific software

- I will just have to do it once
- Nobody cares on how I do get to the answer, I will never share this with others
- If I spare time, I will publish faster

The wrong way of writing scientific software

- I will just have to do it once
- Nobody cares on how I do get to the answer, I will never share this with others
- If I spare time, I will publish faster
- The computer will understand anyway

The wrong way of writing scientific software

- I will just have to do it once
- Nobody cares on how I do get to the answer, I will never share this with others
- If I spare time, I will publish faster
- The computer will understand anyway
- If it works with this input, it works for any input

The wrong way of writing scientific software

- I will just have to do it once
- Nobody cares on how I do get to the answer, I will never share this with others
- If I spare time, I will publish faster
- The computer will understand anyway
- If it works with this input, it works for any input
- The resources I have is what everybody have

The wrong way of writing scientific software

- I will just have to do it once
- Nobody cares on how I do get to the answer, I will never share this with others
- If I spare time, I will publish faster
- The computer will understand anyway
- If it works with this input, it works for any input
- The resources I have is what everybody have
- The data format is secondary, I know what they are

The wrong way of writing scientific software

- I will just have to do it once
- Nobody cares on how I do get to the answer, I will never share this with others
- If I spare time, I will publish faster
- The computer will understand anyway
- If it works with this input, it works for any input
- The resources I have is what everybody have
- The data format is secondary, I know what they are
- The precision I need in result is the one the computer uses

The wrong way of writing scientific software

- I will just have to do it once
- Nobody cares on how I do get to the answer, I will never share this with others
- If I spare time, I will publish faster
- The computer will understand anyway
- If it works with this input, it works for any input
- The resources I have is what everybody have
- The data format is secondary, I know what they are
- The precision I need in result is the one the computer uses
- Software professionals are just overstimated freaks

The Scientist as a programmer

- Poor knowledge of the Hardware

The Scientist as a programmer

- Poor knowledge of the Hardware
- Poor knowledge of the System Software

The Scientist as a programmer

- Poor knowledge of the Hardware
- Poor knowledge of the System Software
- Poor knowledge of the Tools

The Scientist as a programmer

- Poor knowledge of the Hardware
- Poor knowledge of the System Software
- Poor knowledge of the Tools
- Usually little money but lot of manpower

The Scientist as a programmer

- Poor knowledge of the Hardware
- Poor knowledge of the System Software
- Poor knowledge of the Tools
- Usually little money but lot of manpower
- Usually forced to someone else best choices

The Scientist as a programmer

- Poor knowledge of the Hardware
- Poor knowledge of the System Software
- Poor knowledge of the Tools
- Usually little money but lot of manpower
- Usually forced to someone else best choices
- Very conservative, low sharing attitude

Software Engineering

- Design solution under constraint

Software Engineering

- Design solution under constraint
  - Understanding the computer

Software Engineering

- Design solution under constraint
  - Understanding the computer
  - Understanding the problem

Software Engineering

- Design solution under constraint
  - Understanding the computer
  - Understanding the problem
  - Fit the two together and choose the right tools

Software Engineering

- Design solution under constraint
  - Understanding the computer
  - Understanding the problem
  - Fit the two together and choose the right tools
  - Standards in both coding and data

Software Engineering

- Design solution under constraint
  - Understanding the computer
  - Understanding the problem
  - Fit the two together and choose the right tools
  - Standards in both coding and data
  - Documentation of code and algorithm

Software Engineering

- Design solution under constraint
  - Understanding the computer
  - Understanding the problem
  - Fit the two together and choose the right tools
  - Standards in both coding and data
  - Documentation of code and algorithm
  - Test on different platforms

Software Engineering

- Design solution under constraint
  - Understanding the computer
  - Understanding the problem
  - Fit the two together and choose the right tools
  - Standards in both coding and data
  - Documentation of code and algorithm
  - Test on different platforms
  - Check possible failures

Software Engineering

- Design solution under constraint
  - Understanding the computer
  - Understanding the problem
  - Fit the two together and choose the right tools
  - Standards in both coding and data
  - Documentation of code and algorithm
  - Test on different platforms
  - Check possible failures
  - Whenever possible, seek for help

Understand the computer

- Mathematic is declarative (What is knowledge)

Understand the computer

- Mathematic is declarative (What is knowledge)
- Computer Science is imperative (How to knowledge)

Understand the computer

- Mathematic is declarative (What is knowledge)
- Computer Science is imperative (How to knowledge)
  - We need a language to describe process

Understand the computer

- Mathematic is declarative (What is knowledge)
- Computer Science is imperative (How to knowledge)
  - We need a language to describe process
  - We need a logic to prescribe process

Understand the computer

- Mathematic is declarative (What is knowledge)
- Computer Science is imperative (How to knowledge)
  - We need a language to describe process
  - We need a logic to prescribe process
- Floating Point Arithmetic

Understand the computer

- Mathematic is declarative (What is knowledge)
- Computer Science is imperative (How to knowledge)
  - We need a language to describe process
  - We need a logic to prescribe process
- Floating Point Arithmetic
  - Read Dadid Goldberg Paper What every computer scientist should know about floating-point arithmetic

Understand the computer

- Mathematic is declarative (What is knowledge)
- Computer Science is imperative (How to knowledge)
  - We need a language to describe process
  - We need a logic to prescribe process
- Floating Point Arithmetic
  - Read Dadid Goldberg Paper What every computer scientist should know about floating-point arithmetic
- Computer resources: economics of the calculation

Understand the problem

- Think with Data Structures

Understand the problem

- Think with Data Structures
  - Design Data Structures and access functions to them

Understand the problem

- Think with Data Structures
  - Design Data Structures and access functions to them
  - Design the program as a sequence of operations on data

Understand the problem

- Think with Data Structures
  - Design Data Structures and access functions to them
  - Design the program as a sequence of operations on data
  - Design them to be expandable

Understand the problem

- Think with Data Structures
  - Design Data Structures and access functions to them
  - Design the program as a sequence of operations on data
  - Design them to be expandable
- Design a flow diagram of the program and write it down

Understand the problem

- Think with Data Structures
  - Design Data Structures and access functions to them
  - Design the program as a sequence of operations on data
  - Design them to be expandable
- Design a flow diagram of the program and write it down
  - Use standard tools if available (UML)

Understand the problem

- Think with Data Structures
  - Design Data Structures and access functions to them
  - Design the program as a sequence of operations on data
  - Design them to be expandable
- Design a flow diagram of the program and write it down
  - Use standard tools if available (UML)
  - Use anyway the pseudocode

Understand the problem

- Think with Data Structures
  - Design Data Structures and access functions to them
  - Design the program as a sequence of operations on data
  - Design them to be expandable
- Design a flow diagram of the program and write it down
  - Use standard tools if available (UML)
  - Use anyway the pseudocode
- Evaluate the requirements for the execution

Why You should follow coding standard

- You can understand Your own programs

Why You should follow coding standard

- You can understand Your own programs
- Someone else can understand Your own programs

Why You should follow coding standard

- You can understand Your own programs
- Someone else can understand Your own programs
- You can have external contribution You understand

Why You should follow coding standard

- You can understand Your own programs
- Someone else can understand Your own programs
- You can have external contribution You understand
- You spare time in the long run: Science must be reproducible

Why You should use a revision control system

- You can follow the development of the program

Why You should use a revision control system

- You can follow the development of the program
- You can develop the program in a team

Why You should use a revision control system

- You can follow the development of the program
- You can develop the program in a team
- You can revert the modifications done at any time

Why You should use a revision control system

- You can follow the development of the program
- You can develop the program in a team
- You can revert the modifications done at any time
- You can calculate easily efforts required

Why You should plan testing

- You should test Your program for valid and invalid input

Why You should plan testing

- You should test Your program for valid and invalid input
- You want to have a way to test it again in the future

Why You should plan testing

- You should test Your program for valid and invalid input
- You want to have a way to test it again in the future
- You may want to find problems and bugs

Why You should plan testing

- You should test Your program for valid and invalid input
- You want to have a way to test it again in the future
- You may want to find problems and bugs
- You want simple tools to follow program flow

Why You should have a Software Forge

- What is a Collaborative Development Environment?

Why You should have a Software Forge

- What is a Collaborative Development Environment?
  - Web based collaboration site

Why You should have a Software Forge

- What is a Collaborative Development Environment?
  - Web based collaboration site
  - Code repository and project management

Why You should have a Software Forge

- What is a Collaborative Development Environment?
  - Web based collaboration site
  - Code repository and project management
  - Code maintenance and versioning

Why You should have a Software Forge

- What is a Collaborative Development Environment?
  - Web based collaboration site
  - Code repository and project management
  - Code maintenance and versioning
  - Documentation platform

Why You should have a Software Forge

- What is a Collaborative Development Environment?
  - Web based collaboration site
  - Code repository and project management
  - Code maintenance and versioning
  - Documentation platform
  - Bugs tracking and ticketing

Scientific codebases

- Is writing from scratch always worth the task?

Scientific codebases

- Is writing from scratch always worth the task?
- Mix of public domain, free software and commercial solutions

Scientific codebases

- Is writing from scratch always worth the task?
- Mix of public domain, free software and commercial solutions
- Commercial environments and tools for research: more the possible outcome, more the commercial

Scientific codebases

- Is writing from scratch always worth the task?
- Mix of public domain, free software and commercial solutions
- Commercial environments and tools for research: more the possible outcome, more the commercial
- Which policy should be used to develop scientific software?

Thank You !

Thank You !