Software Development CYCLE in Astronomy

Eszter Pozna
ESO
Vienna Dynamics Workshop 8-11 October 2019

What is software engineering?

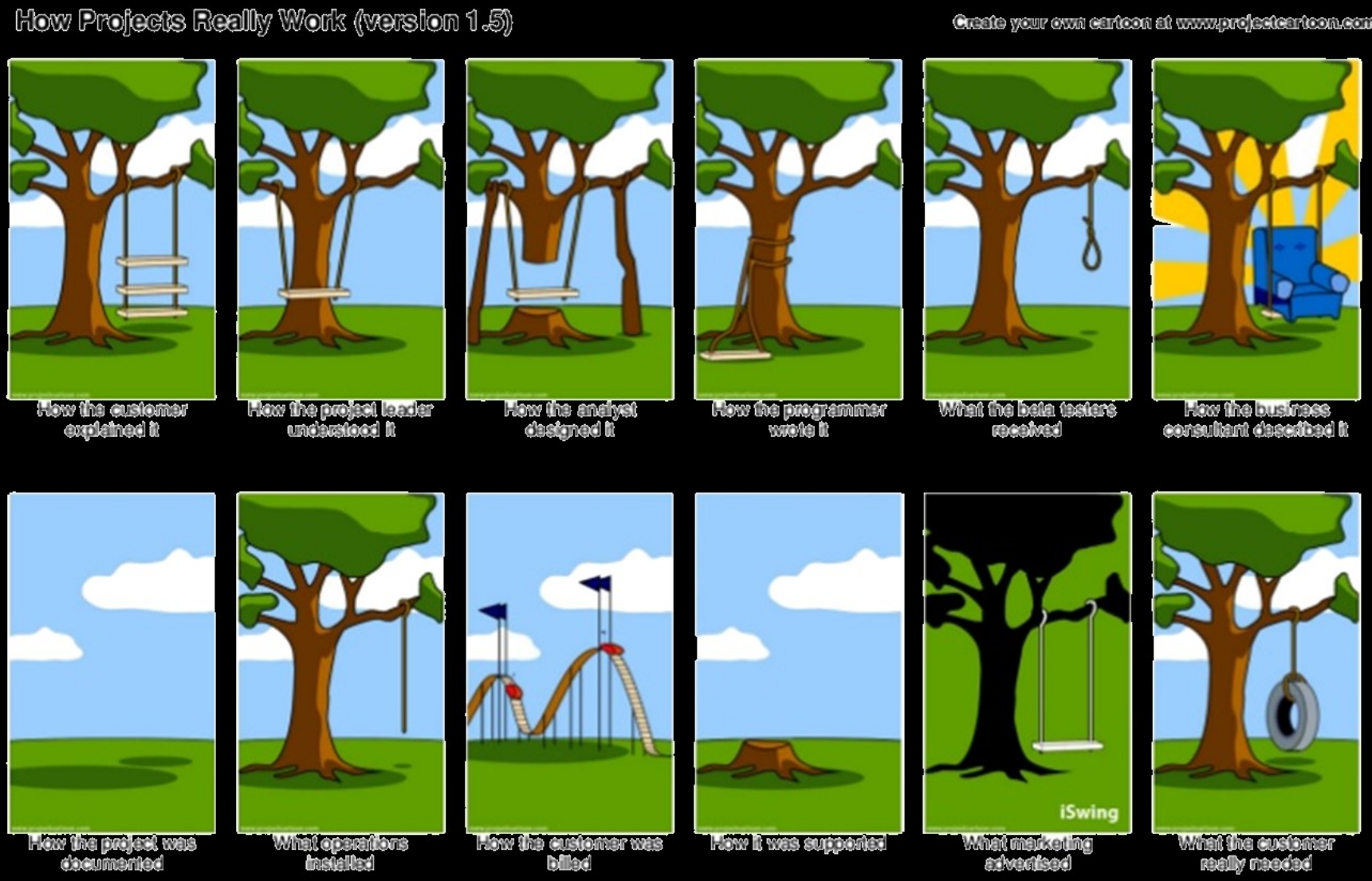
- *Convert ideas to software*
- *Deliver on time*
- Keep it running..(reliably, efficiently)

Its about…

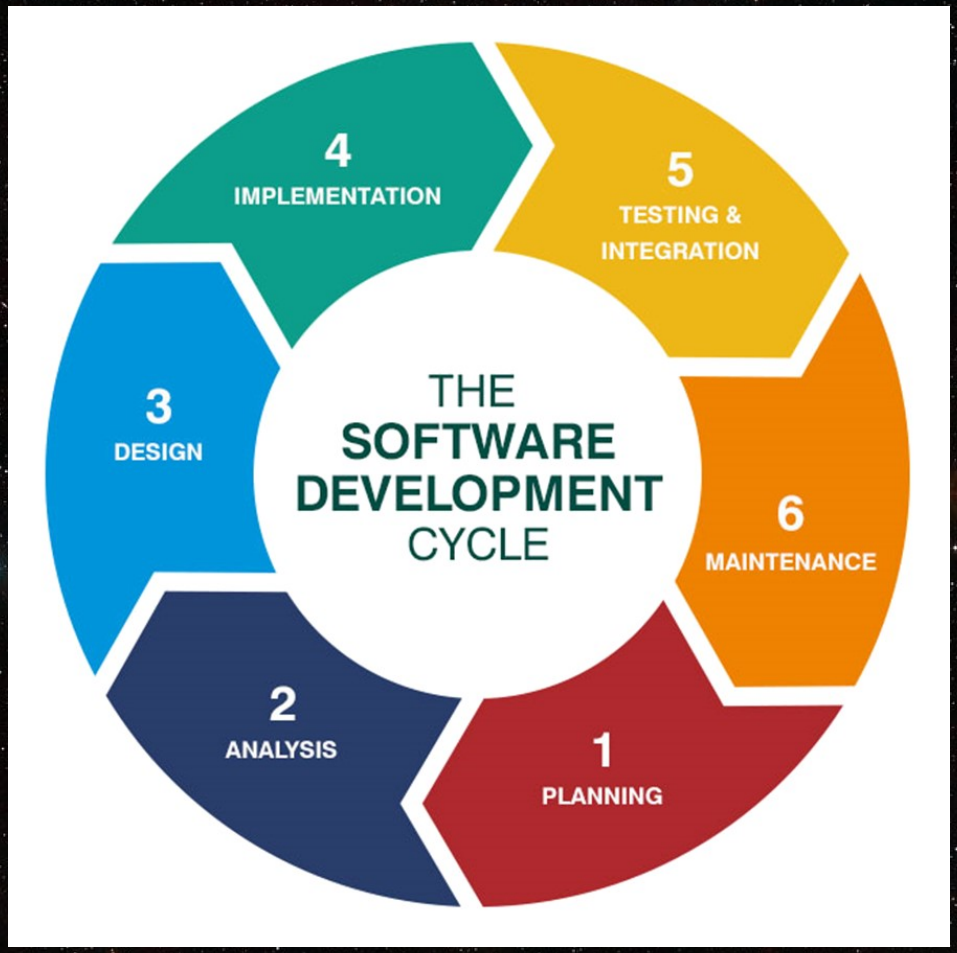
$$m_i \frac{d\mathbf{r}_i}{dt} = \sum_{\substack{i=1 \\ i \neq j}}^{N} \frac{G m_i m_j (\mathbf{r}_i - \mathbf{r}_j)}{\|\mathbf{r}_i - \mathbf{r}_j\|^3}$$

101010

Project pitfalls…



SDLC

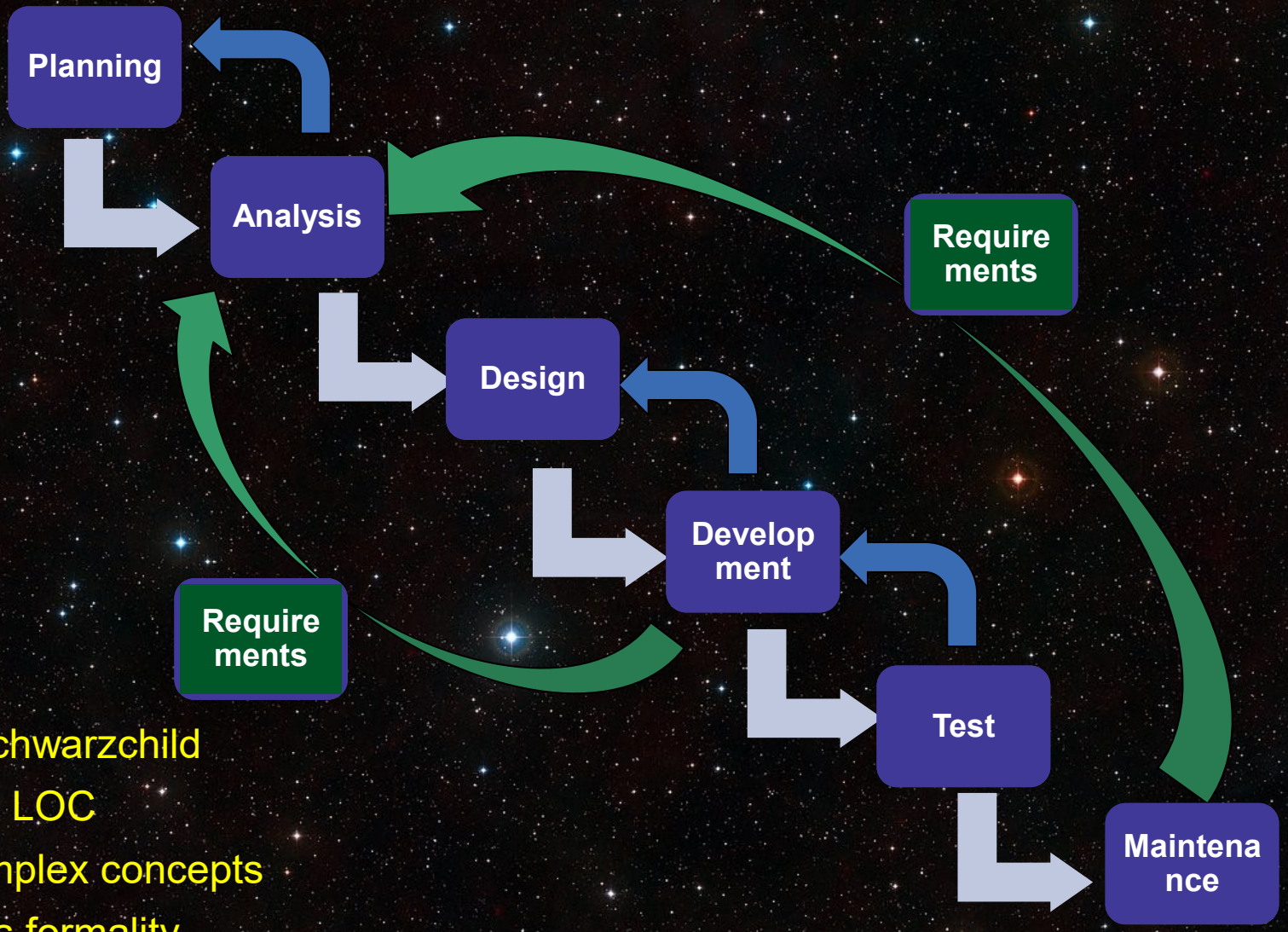

4
IMPLEMENTATION
5
TESTING &
INTEGRATION
3
DESIGN
THE
SOFTWARE
DEVELOPMENT
CYCLE
6
MAINTENANCE
2
ANALYSIS
1
PLANNING

Process for Small project

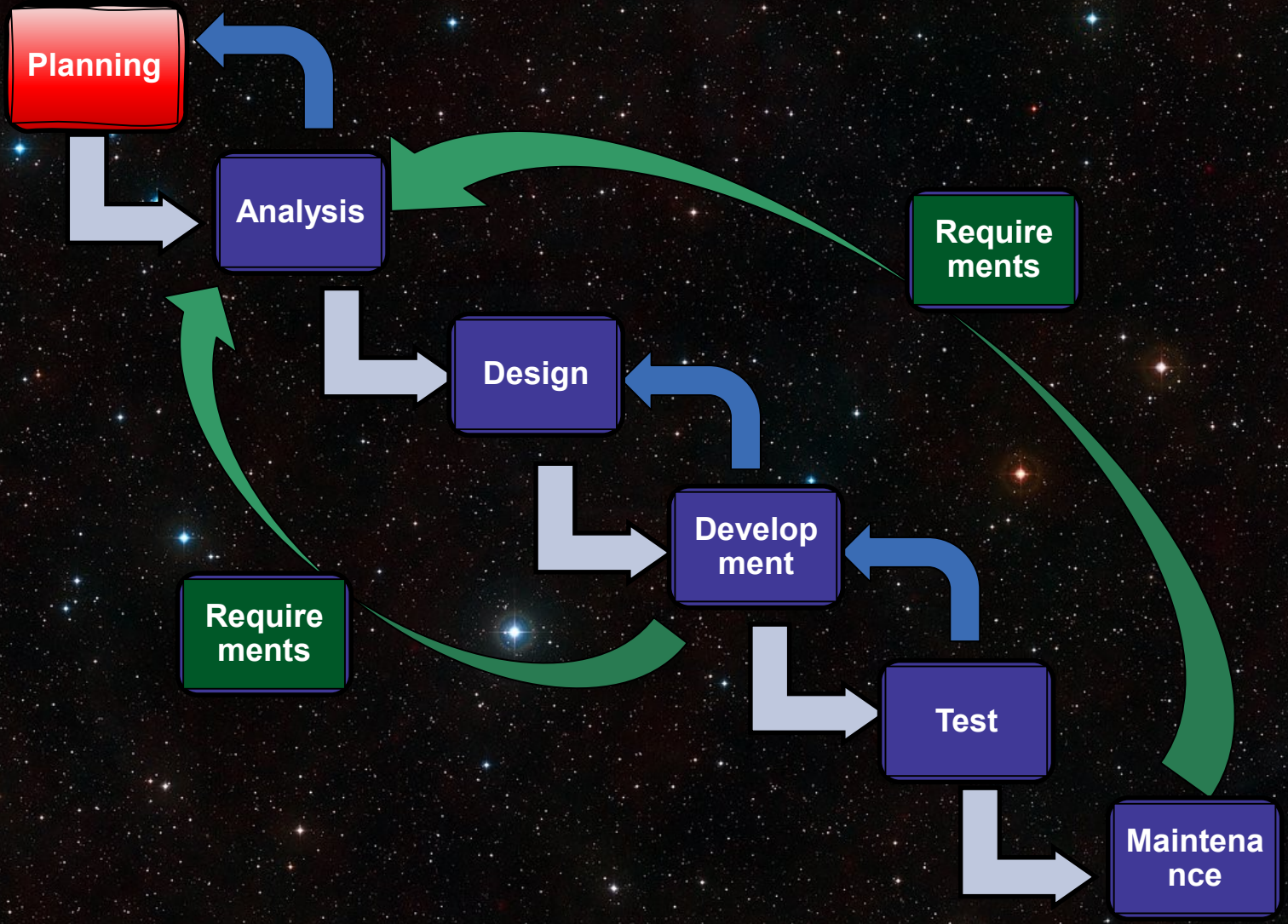
Big Bang Model

- This model is ideal for **small projects like academic projects** or practical projects. One or two developers can work together on this model.
- In this model, developers do not follow any specific process. Development begins with the necessary funds and efforts in the form of inputs.
- And the result may or may not be as per the customer's requirement, because in this model, even the **customer requirements are not defined**.
- https://www.javatpoint.com/software-engineering-big-bang-model

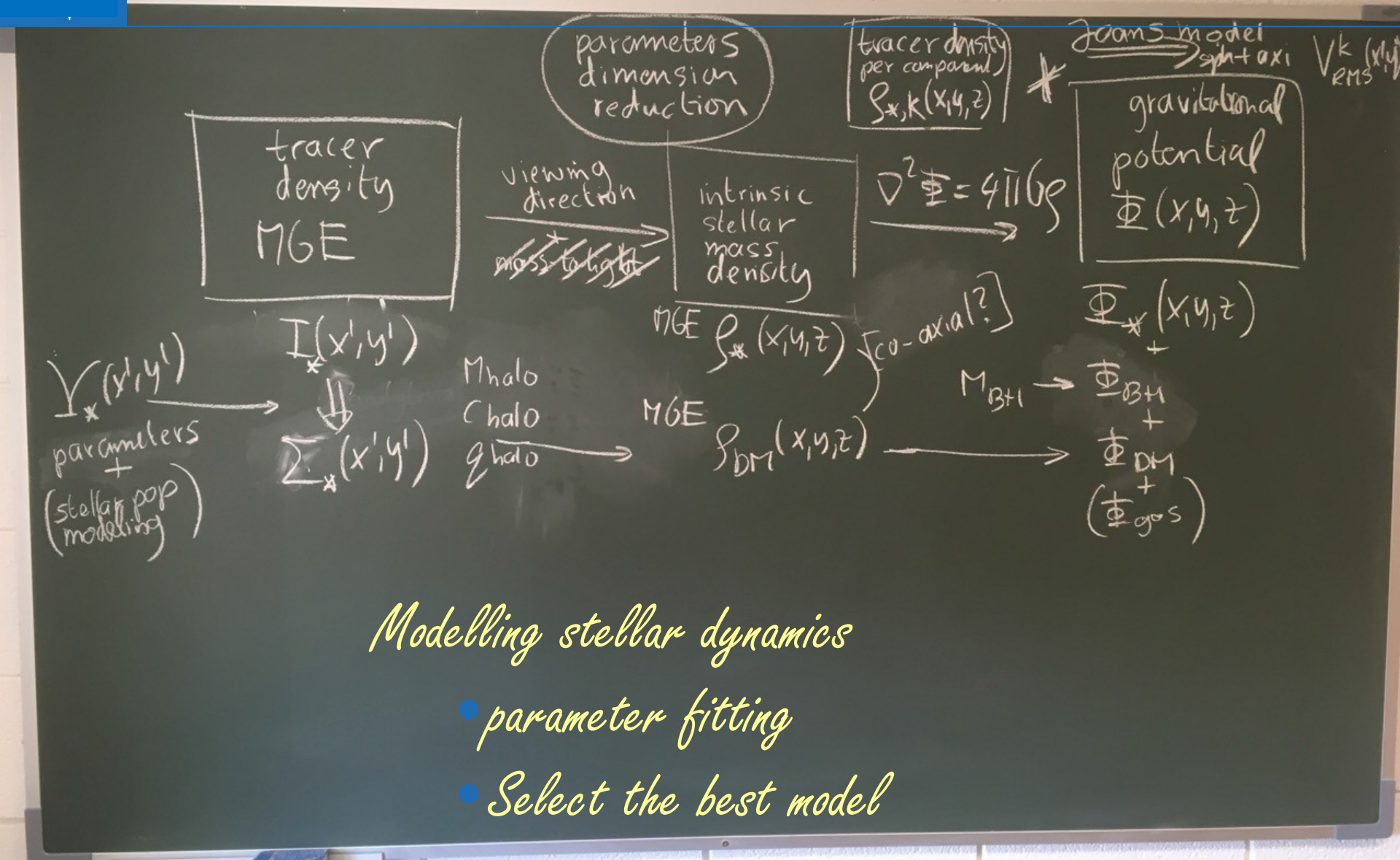
Prototyping



- triaxalSchwarzchild
  - 10K LOC
  - Complex concepts
  - Less formality


Planning
Analysis
Require ments
Design
Develop ment
Require ments
Test
Maintena nce

What?

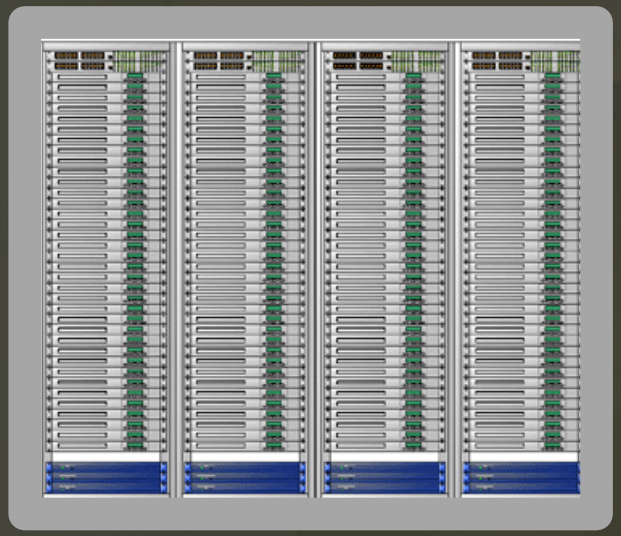


Modelling stellar dynamics

- parameter fitting
- Select the best model

How?

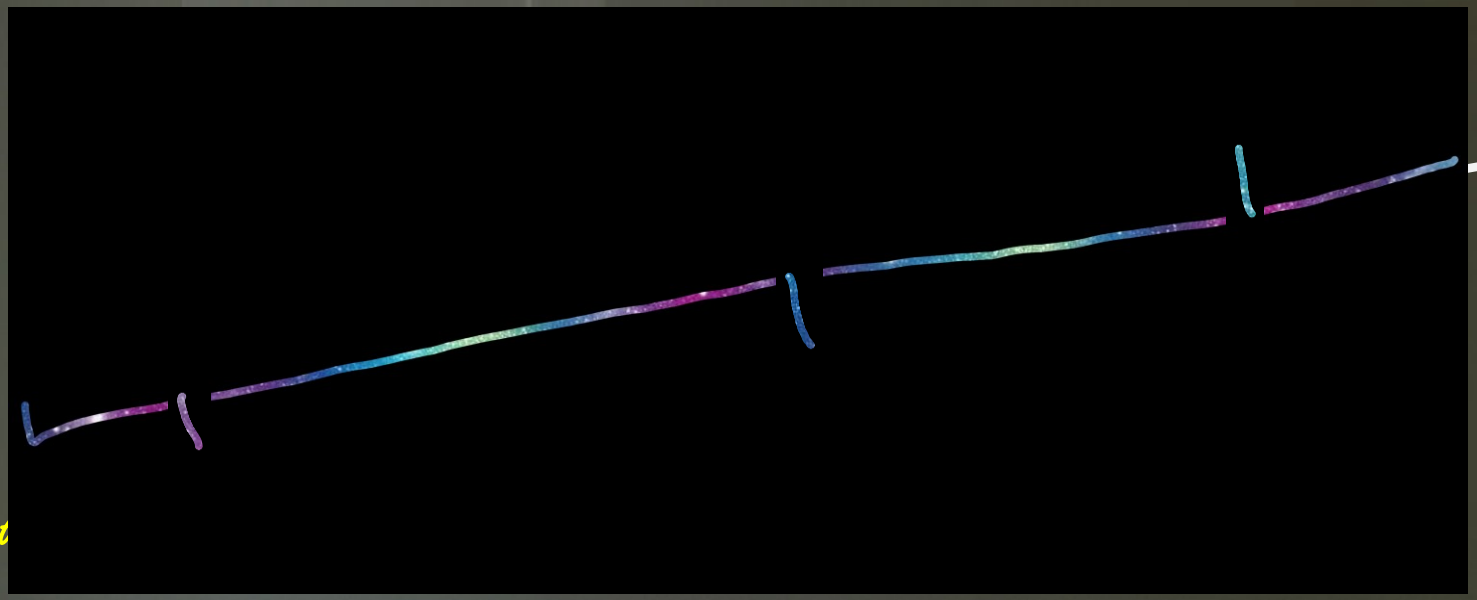
make better use of
underlying parallel hardware

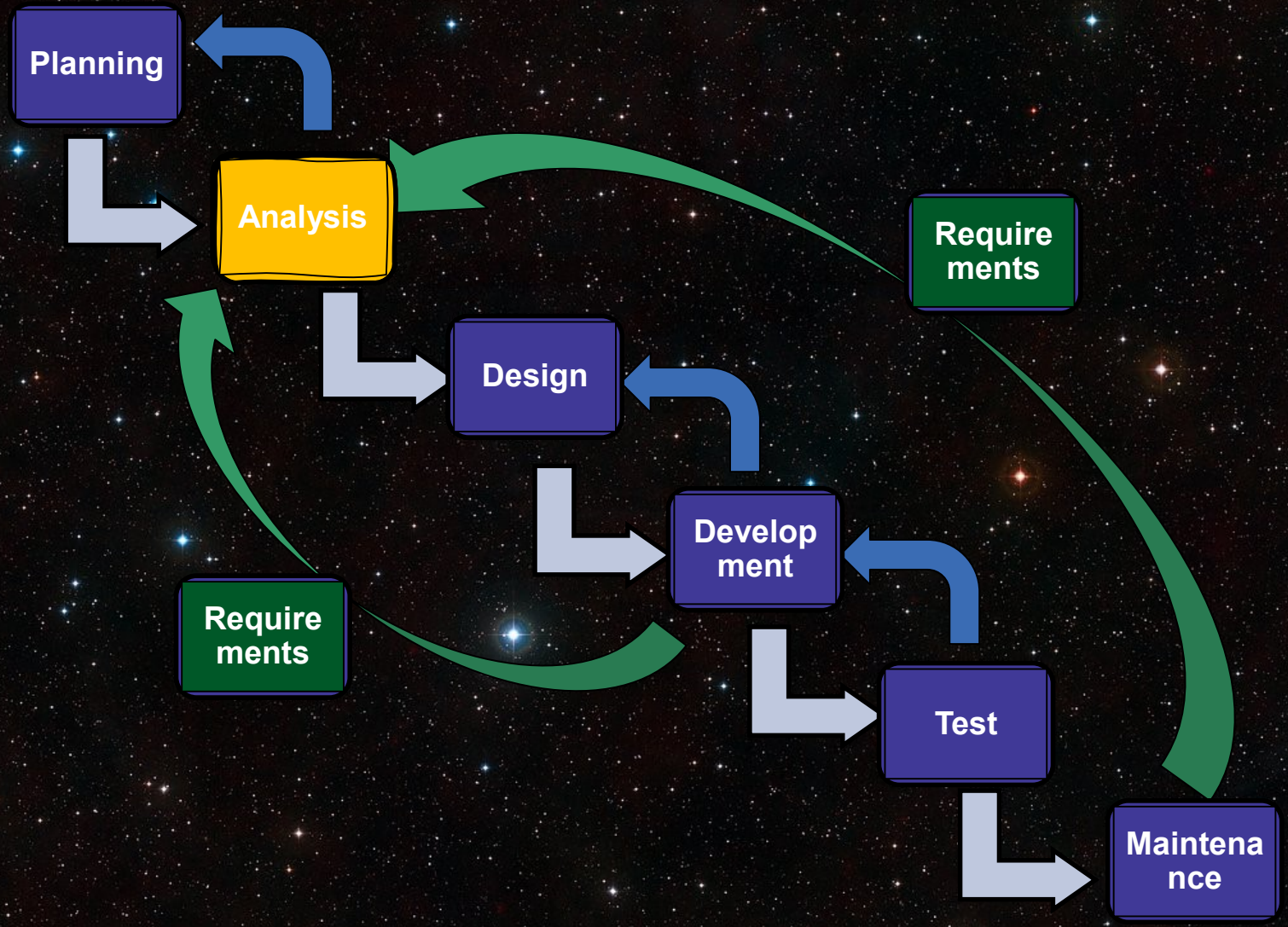


Cluster of computer

Parallel computing is perfect
for modeling, simulating

When?



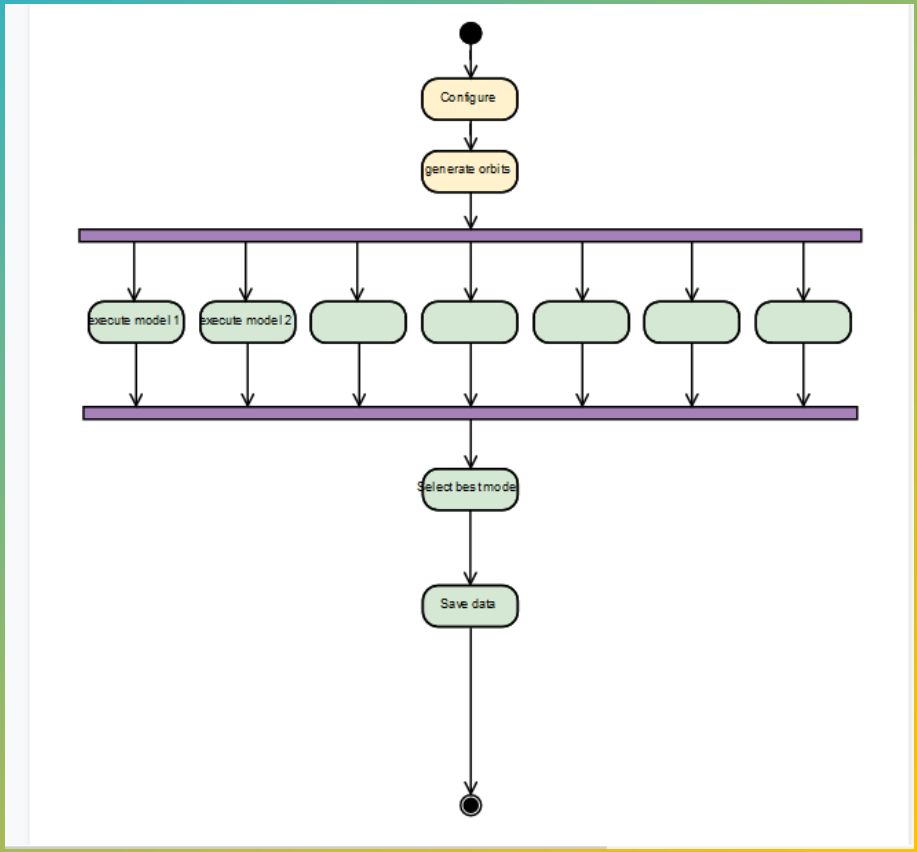

Planning
Analysis
Require ments
Design
Develop ment
Require ments
Test
Maintena nce

Analysis

- Feasibility study
  - Details
  - Prototyping/Testing
  - Discuss GUI
- Check availability reusability of existing software
- Cost, Time

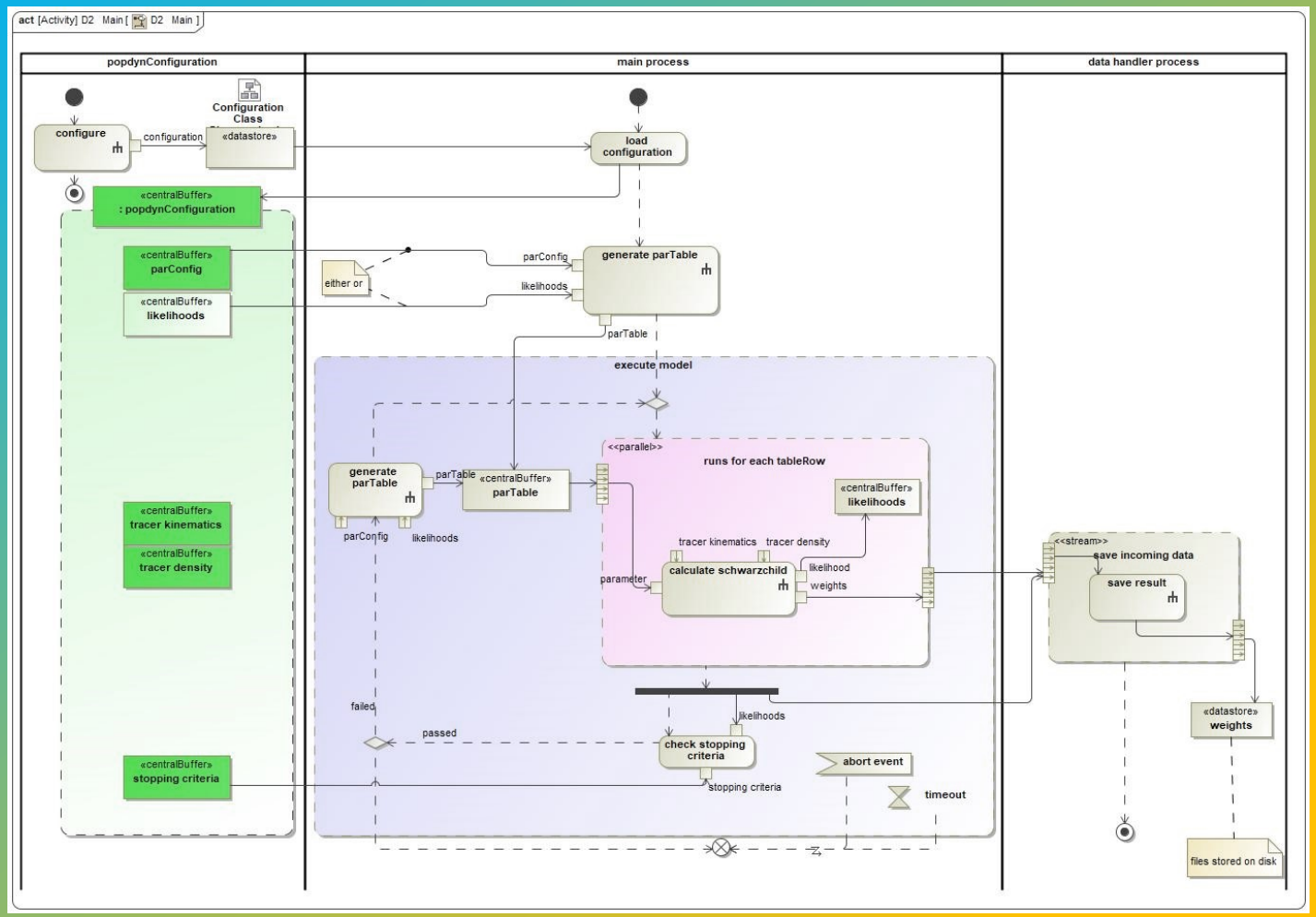
Analysis

- Document your idea in a way that is meaningful for software people (UML, Activity diagram)

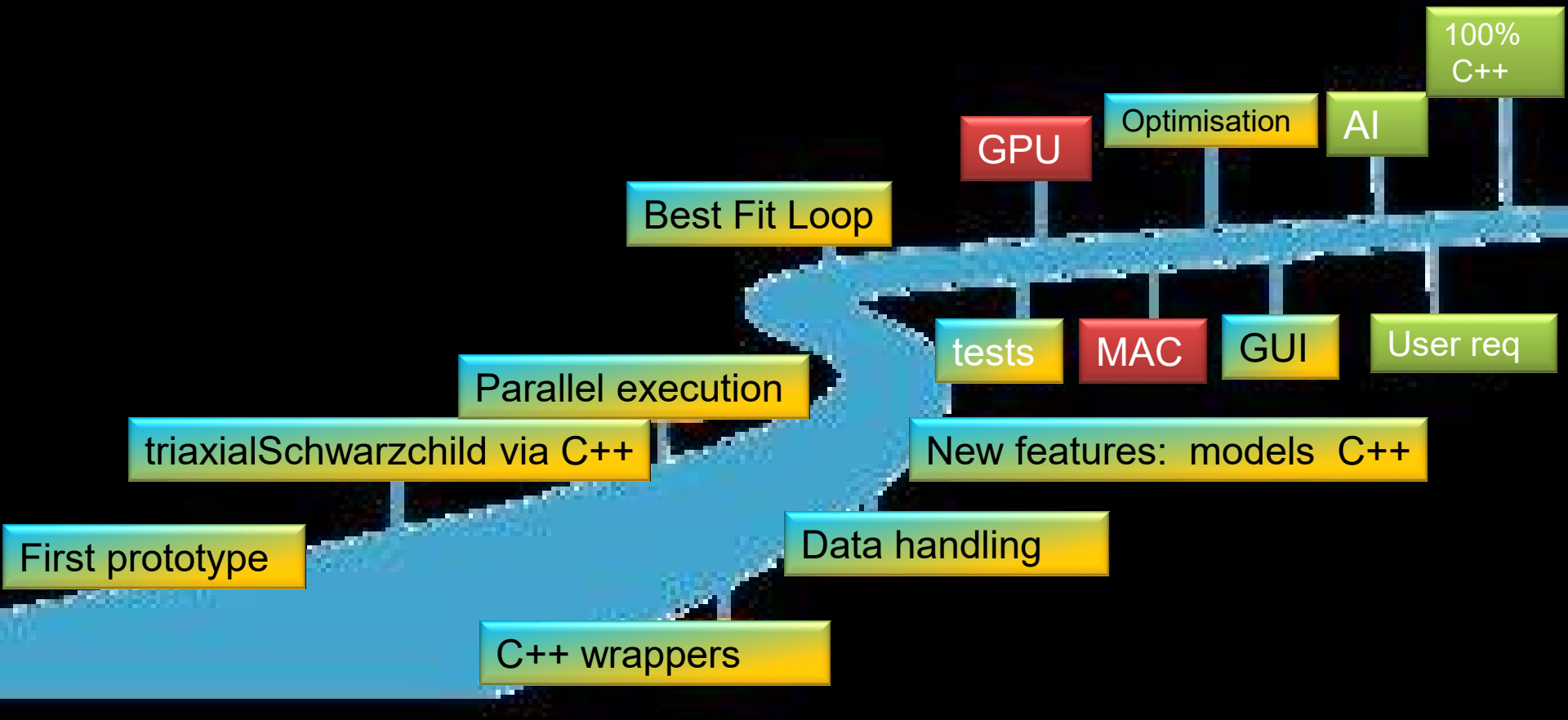


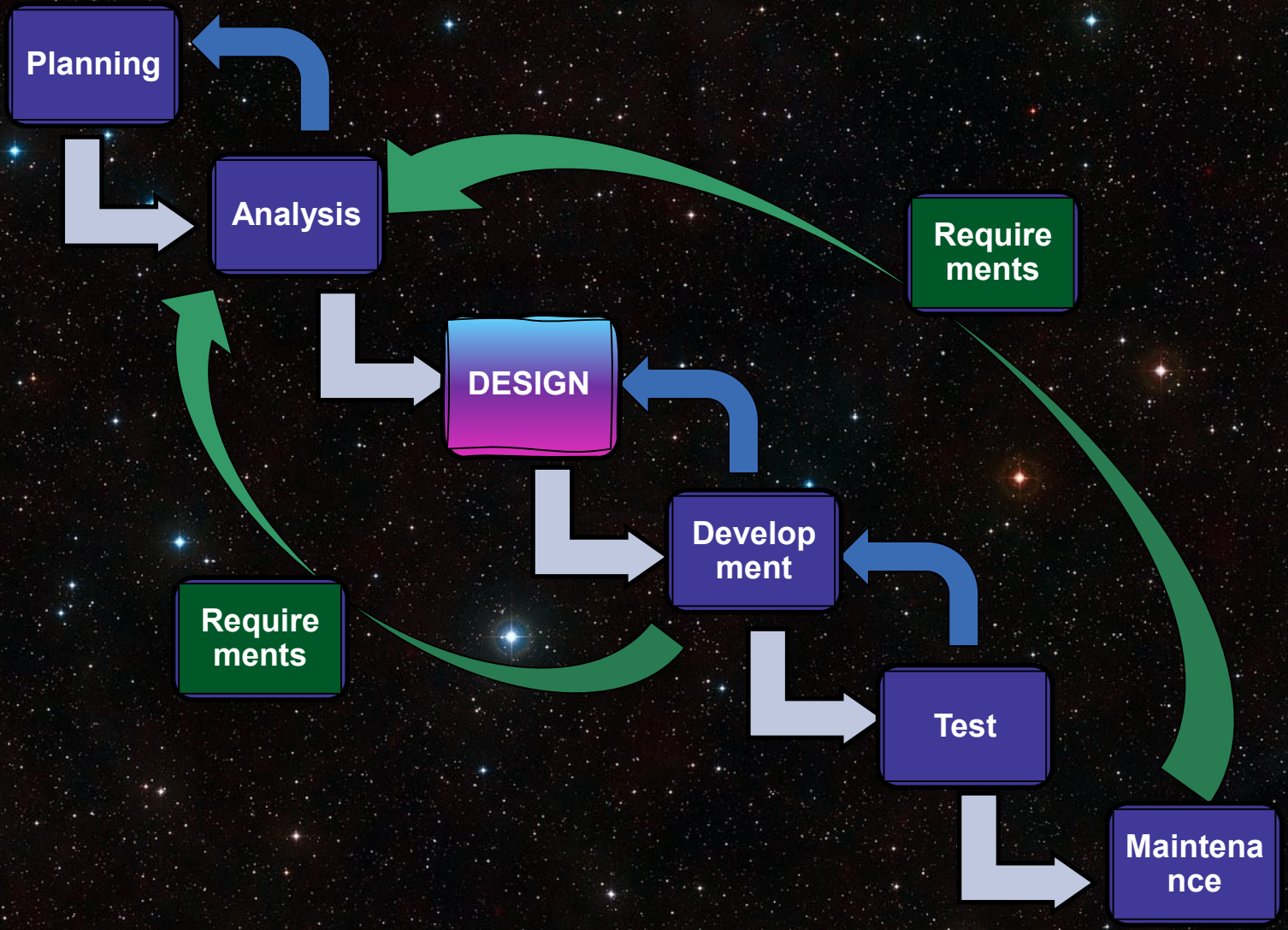
Analysis

- Concept via UML activity diagrams



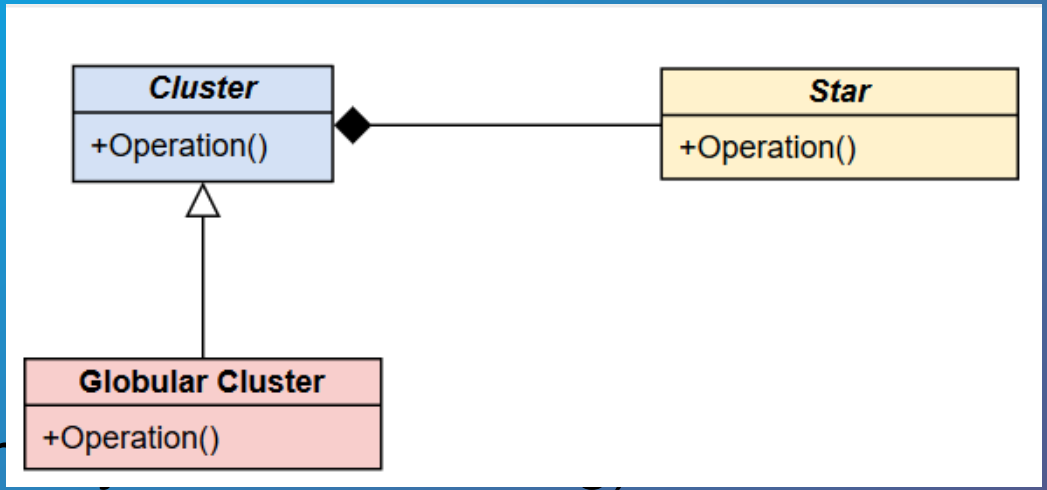
Milestones


100% C++
Optimisation
AI
GPU
Best Fit Loop
tests
MAC
GUI
User req
Parallel execution
triaxialSchwarzchild via C++
New features: models C++
First prototype
Data handling
C++ wrappers


Planning
Analysis
Require ments
DESIGN
Develop ment
Require ments
Test
Maintena nce

Design

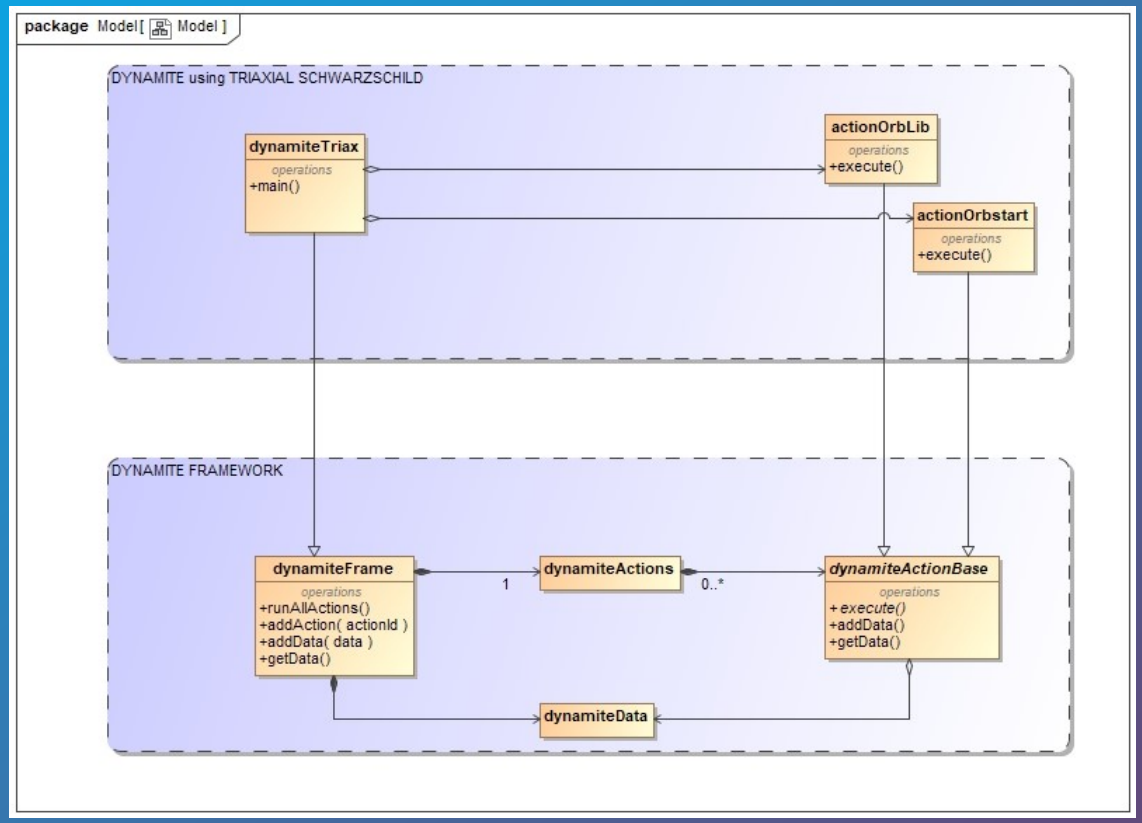
- Architecture
  - Packages
    - classes
  - Connections

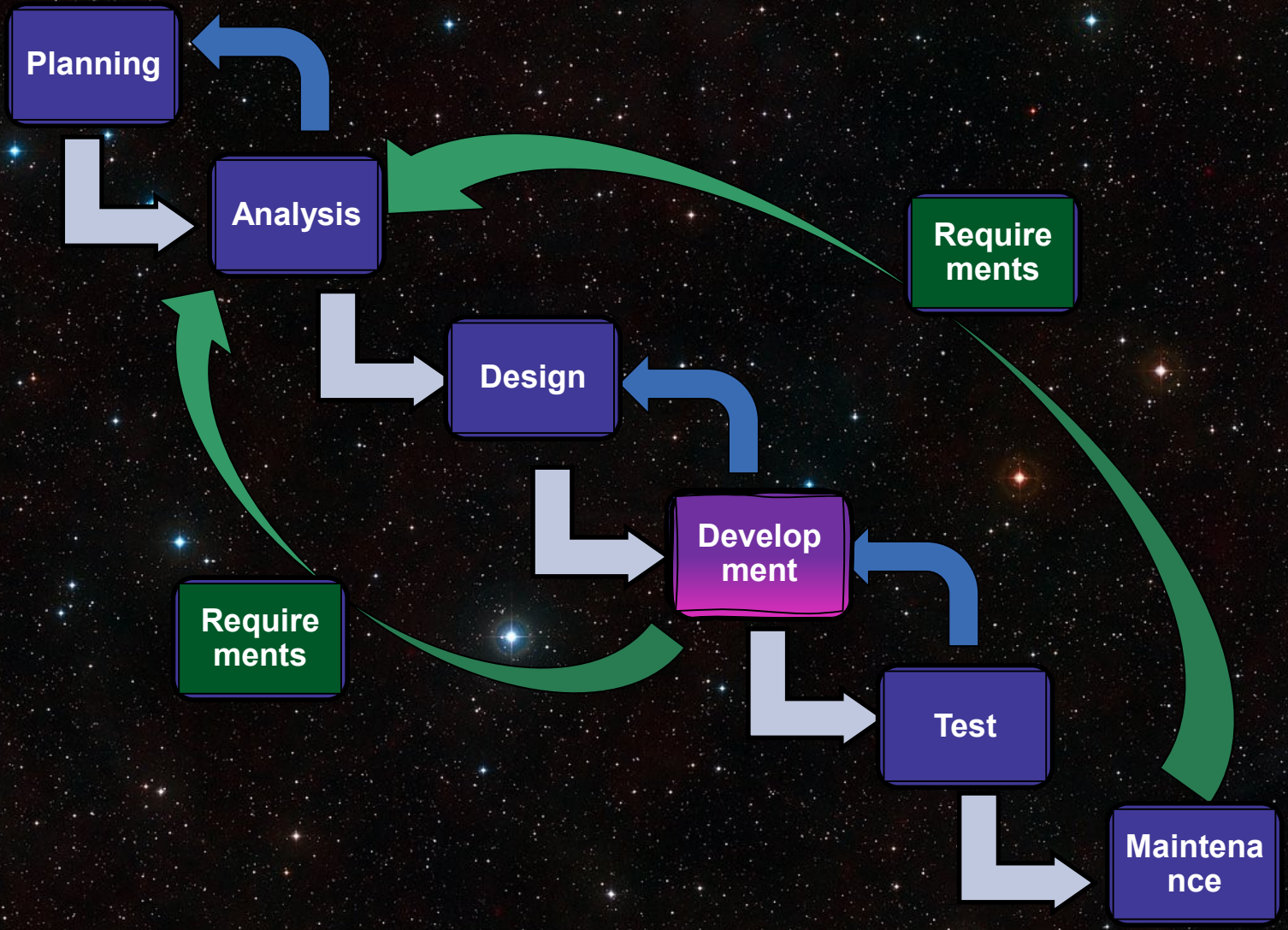


- Data handling (mer
- Flexibility/ Configuration etc.
- Logging and debugging
- User interface

Software Design

- Design patterns
- Design decisions (e.g. process or thread)



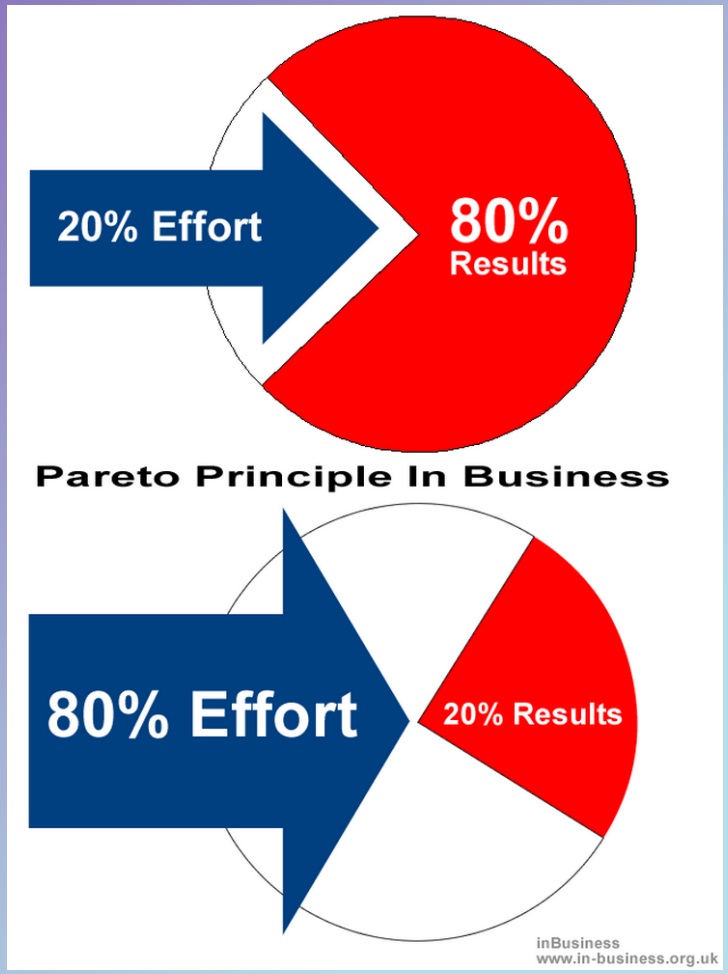

Planning
Analysis
Require ments
Design
Develop ment
Require ments
Test
Maintena nce

Preparation

- Free stuff
  - Platform : Linux by Linus Torvalds
  - GIT version control
  - Language : C++ by Bjarne Stroustroup
  - IDE
  - Download support Libraries
  - Installation method (Makefile,Maven,Gradel,Waf..)
  - Problem reporting system (Jira, Bugzilla)
  - Setup Database

Development…

- **80 % of the time the software runs in 20% of the code (Pareto rule)**
  - Stability is given by the rest!!
    - Error handling, Special cases,
    - preparation, logging,
    - startup/shutdown actions
- Unit test
- Documentation



Software development practices

DRY

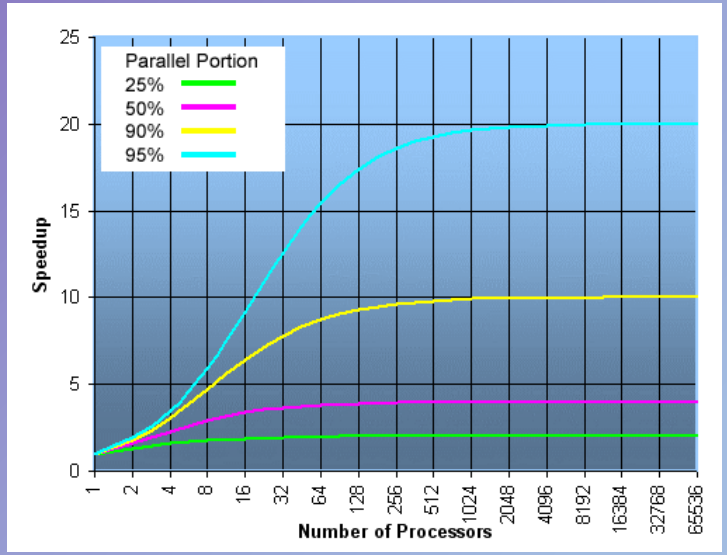
KISS

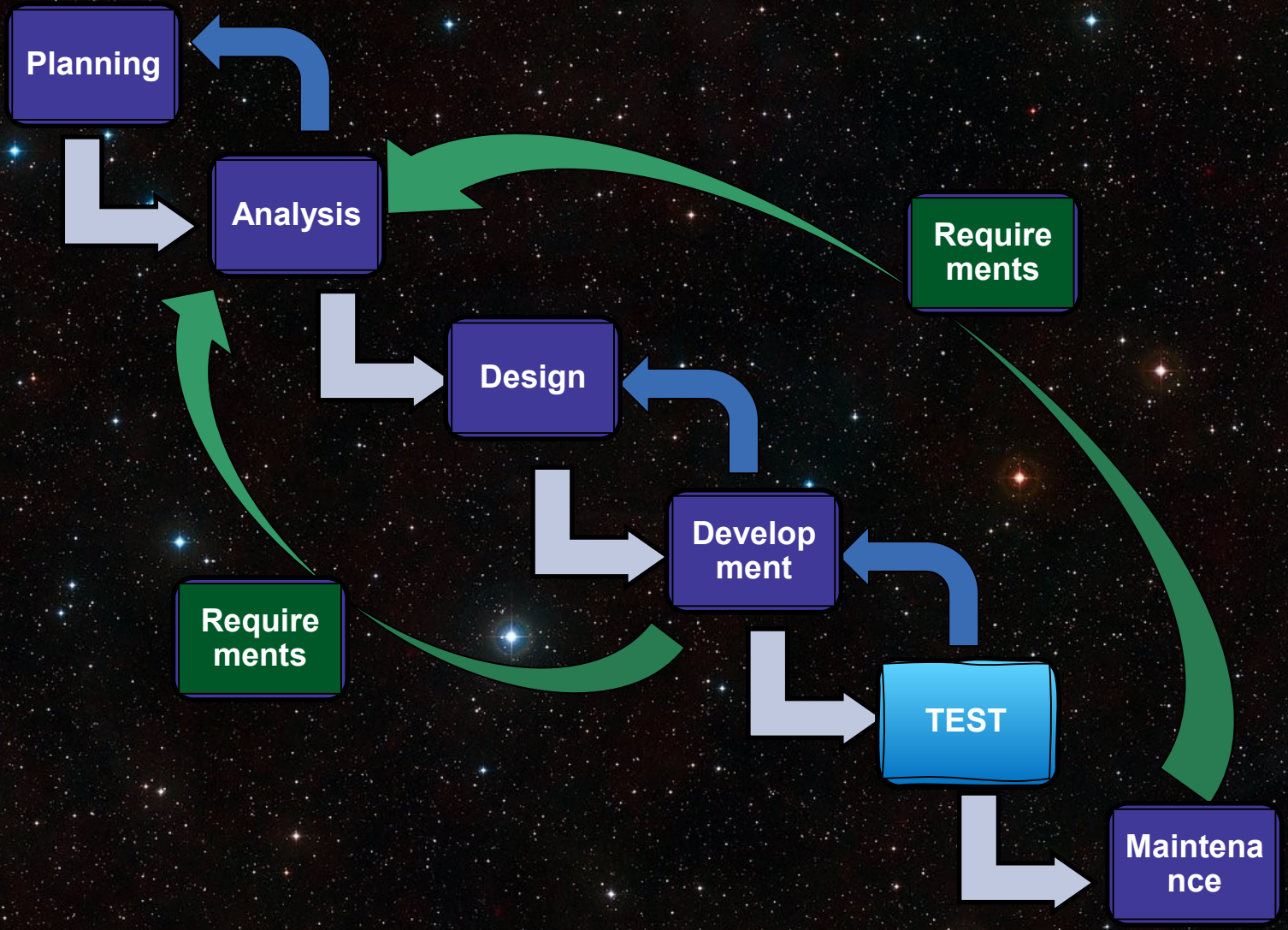
robust

modular

Software Challenges

- Interface Fortran/C++
- Parallelization
  - Race conditions
- Memory handling
- Flexibility via design pattern
- Stability
- Optimization for speed



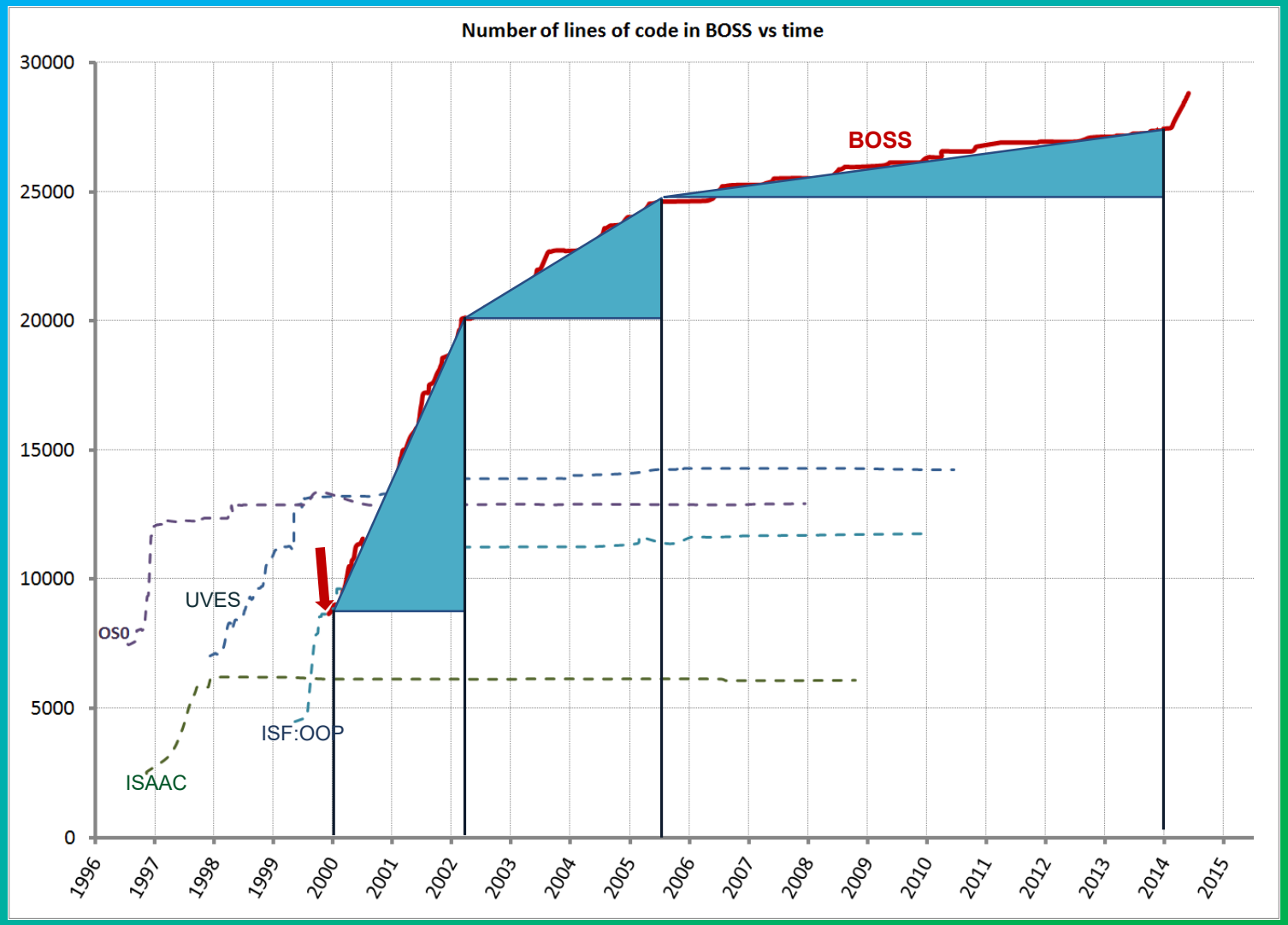

Planning
Analysis
Require ments
Design
Develop ment
Require ments
TEST
Maintena nce

Test and integration

- Test: phaseA (developer) → phaseB (professional testers) → phaseC (requester)
- Integration: Test platform ➔ User platform

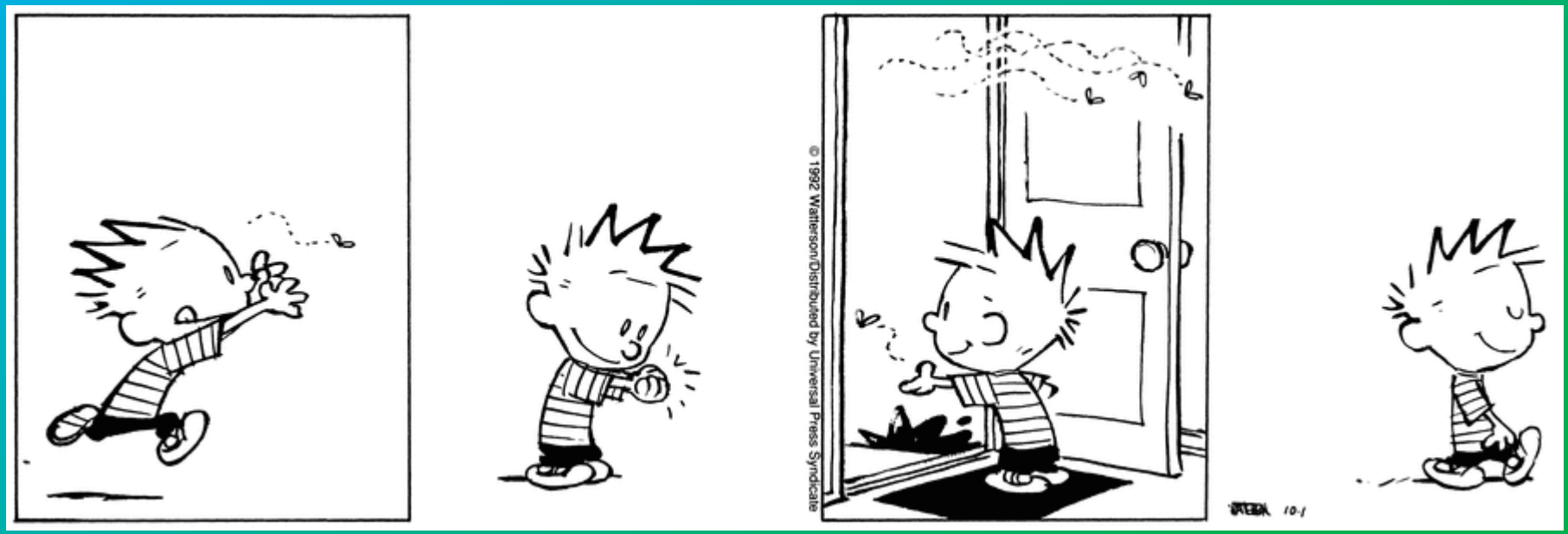
MAINTENANCE

Evolution


Number of lines of code in BOSS vs time
30000
25000
20000
15000
10000
5000
0
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
BOSS
UVES
OS0
ISF:OOP
ISAAC

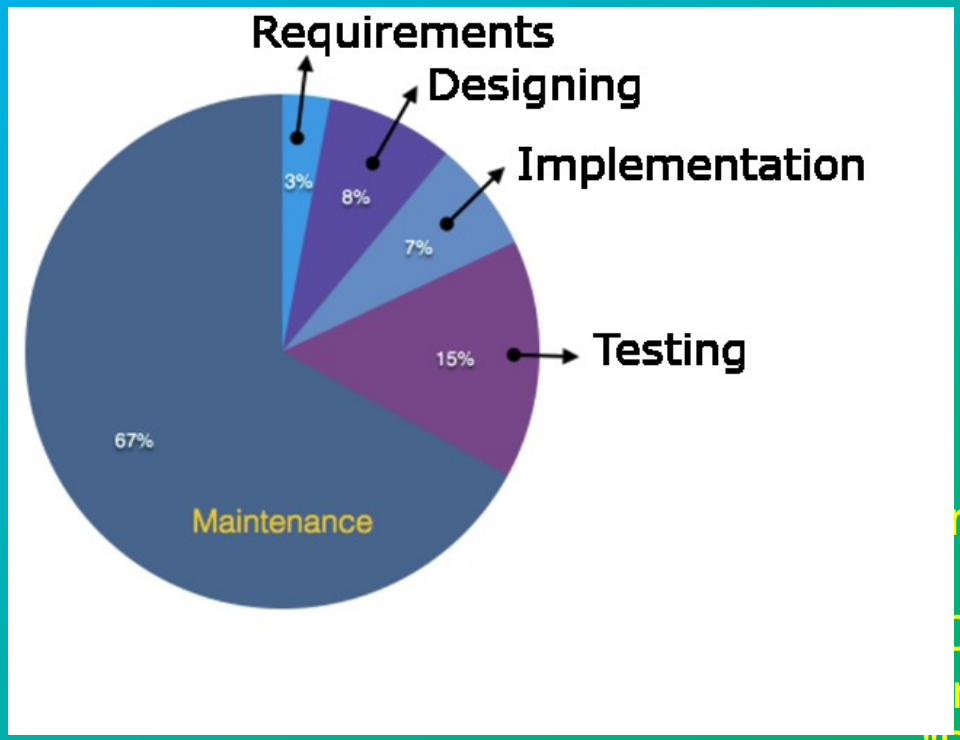
Maintenance

- **repeatedly updating software for various reasons**
  - improvement: Involve changes in functionality
  - Adaptative: Changes in the environment are adapted to the requirements
  - Corrective: Activities for error correction
  - Preventive: Improvements to avoid future problems



MAINTENANCE COST

- 50% and 70% of all total costs



man's Laws

- Continuing Change“
- ncreasing Complexity“
- "Declining Quality"


4
IMPLEMENTATION
5
TESTING &
INTEGRATION
3
DESIGN
END
6
MAINTENANCE
2
ANALYSIS
1
PLANNING