Solving complex Business IT Problems using FOSS - #TDOSE 2016

Solving complex Business IT Problems using FOSS

NO COMPLEXITY

Version	:	1.0
Date	:	12-11-2016
Status	:	Open for Discussion!
Author	:	Maikel Mardjan (@maikelmardjan)

Growth
No Complexity

(Your) IT projects never fail!

WHOAMI

Name : Maikel Mardjan (Dutch) (Twitter @maikelmardjan)

- Architecture & Design
- 20+ years working within IT Industry
- Master (MSc) Business Studies of University of Groningen
- Master degree (MSc) Electrical Engineering, of Delft University of Technology
- …and still likes to do real hands-on programming (Golang, Python, PHP,JS etc) to make and break things

THE Open GROUP
TOGAF™ 9 Certified

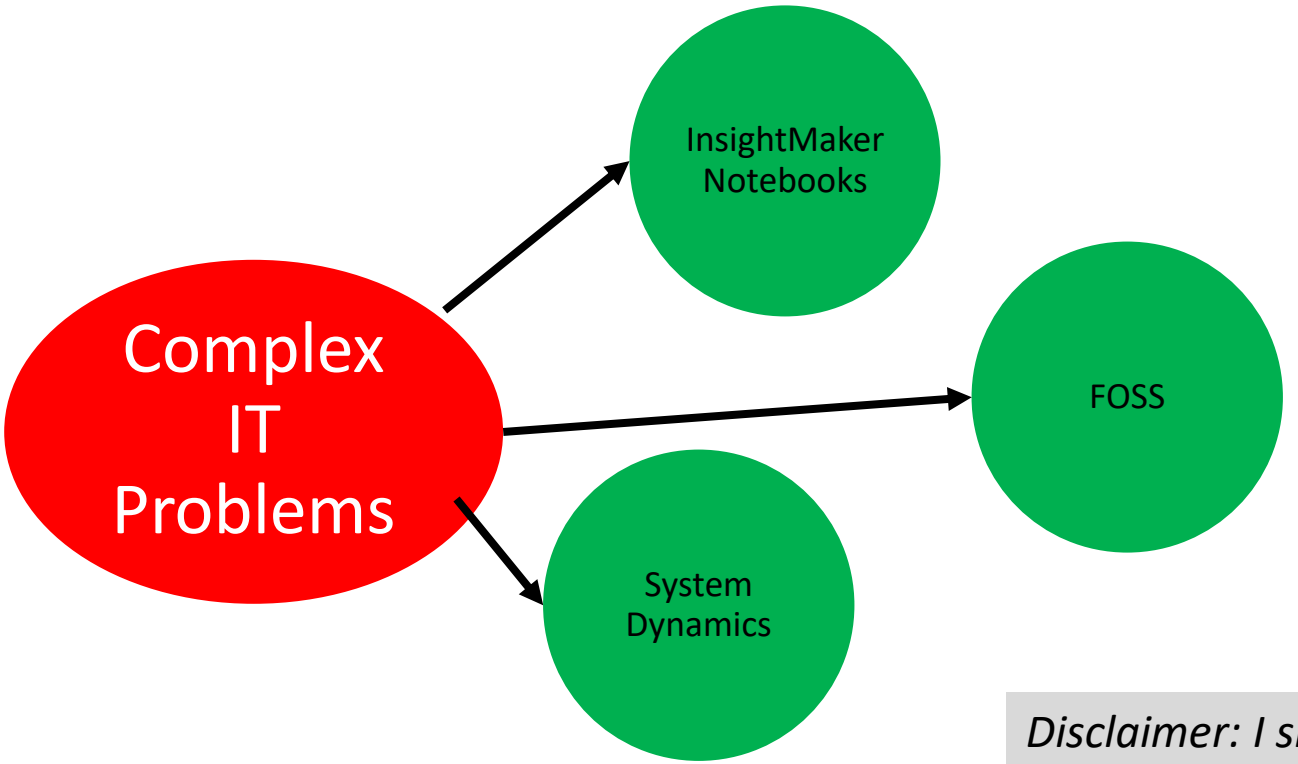
CERTIFIED INFORMATION SYSTEMS SECURITY PROFESSIONAL
CISSP

I love complex IT problem solving!

Growth
NoComplexity.com

Agenda - outline

- IT projects never fail!
- What is system dynamics?
- How to model a complex IT problem?
- FOSS software for system dynamics
- Advantages and disadvantages FOSS tools for problem solving



Disclaimer: I simplify on purpose!

IT PROJECTS NEVER FAIL!

Top software failures 2015: 600,000 RBS payments go missing

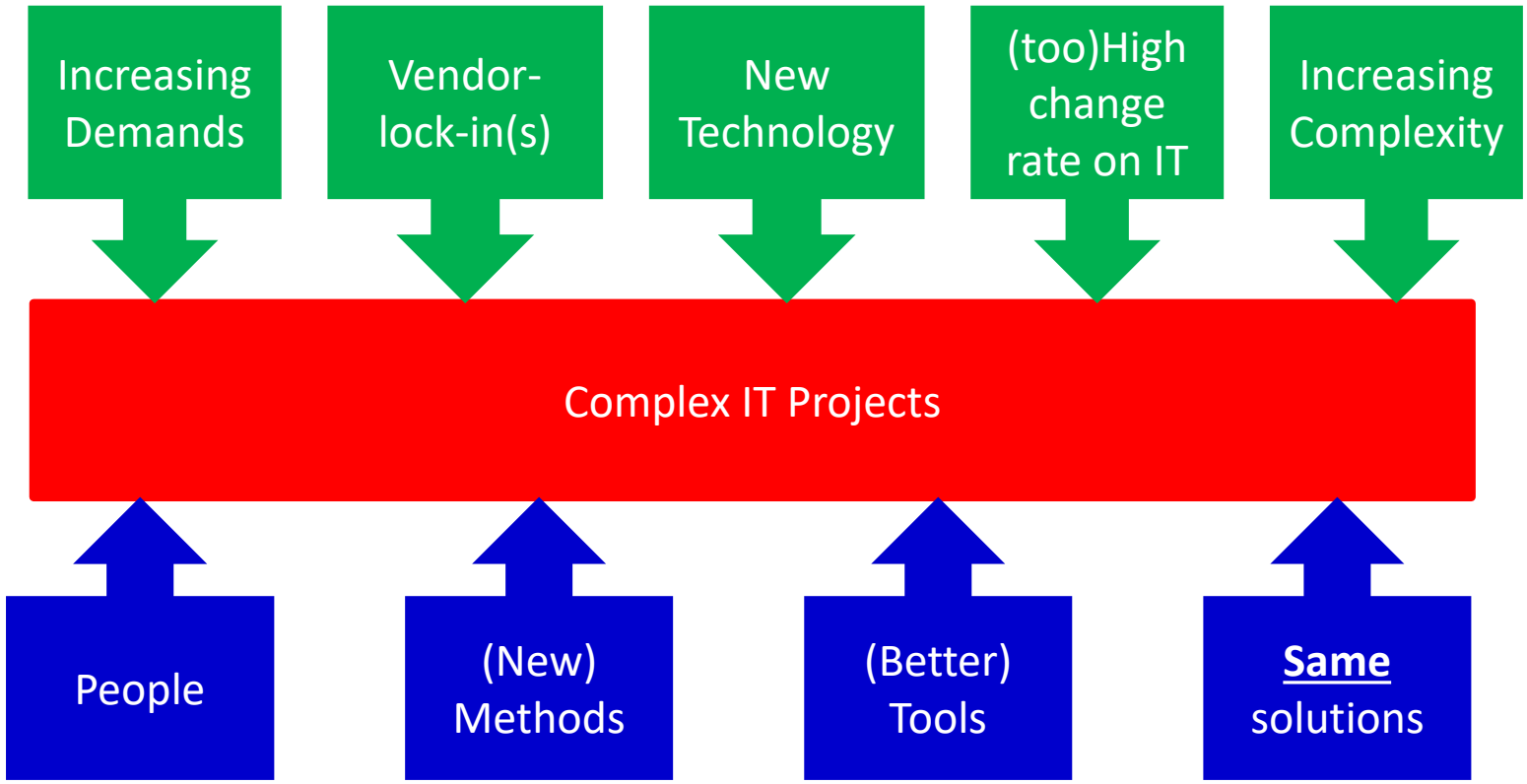
IT PROJECTS NEVER FAIL!

Top software failures 2015: Glitch releases US prisoners early

IT PROJECTS NEVER FAIL!

Top software failures 2014: UK airspace closed

IT PROJECTS NEVER FAIL!



IT PROJECTS NEVER FAIL!

So:

Do not believe anyone that claims to know why projects fail!

(The world is too complex for simple fix lists)

OUTLINE

- IT projects never fail!
- What is system dynamics?
- How to model a complex dynamic IT problem?
- FOSS software for system dynamics
- Advantages and disadvantages FOSS tools for problem solving

System Dynamics

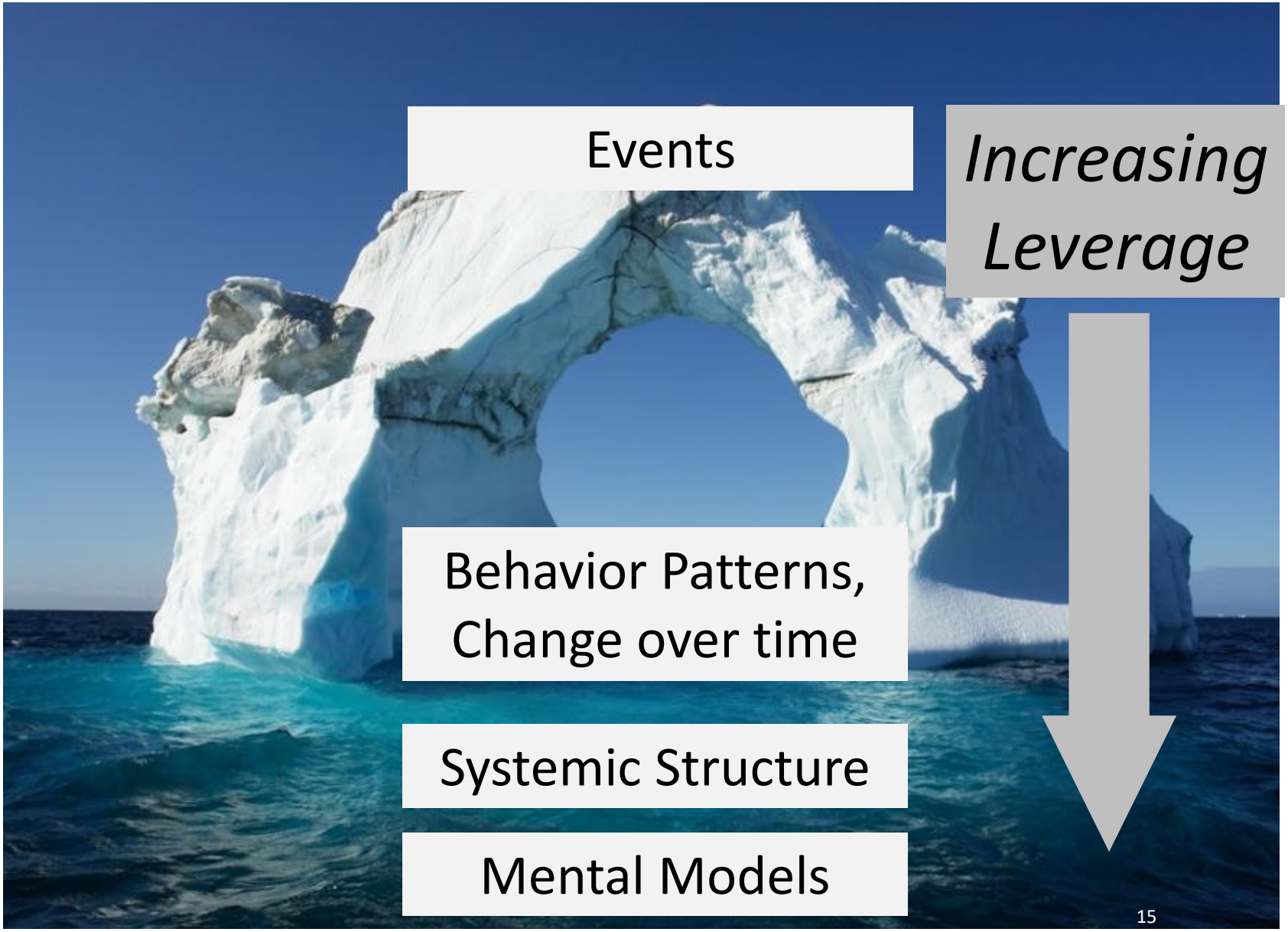
System Dynamics

Known and Unknown behaviour makes IT projects comparable with rocket launching:

- Programming (mistakes)
- Knowledge
- Floating specs
- Budget
- Scope creep
- Software errors
- Hardware errors (CPU, Storage, etc)
- Communication errors (network and humans)
- Human behaviour
- Factor time

System Dynamics

System Dynamics



Most Nasty Factors that makes IT problems complex

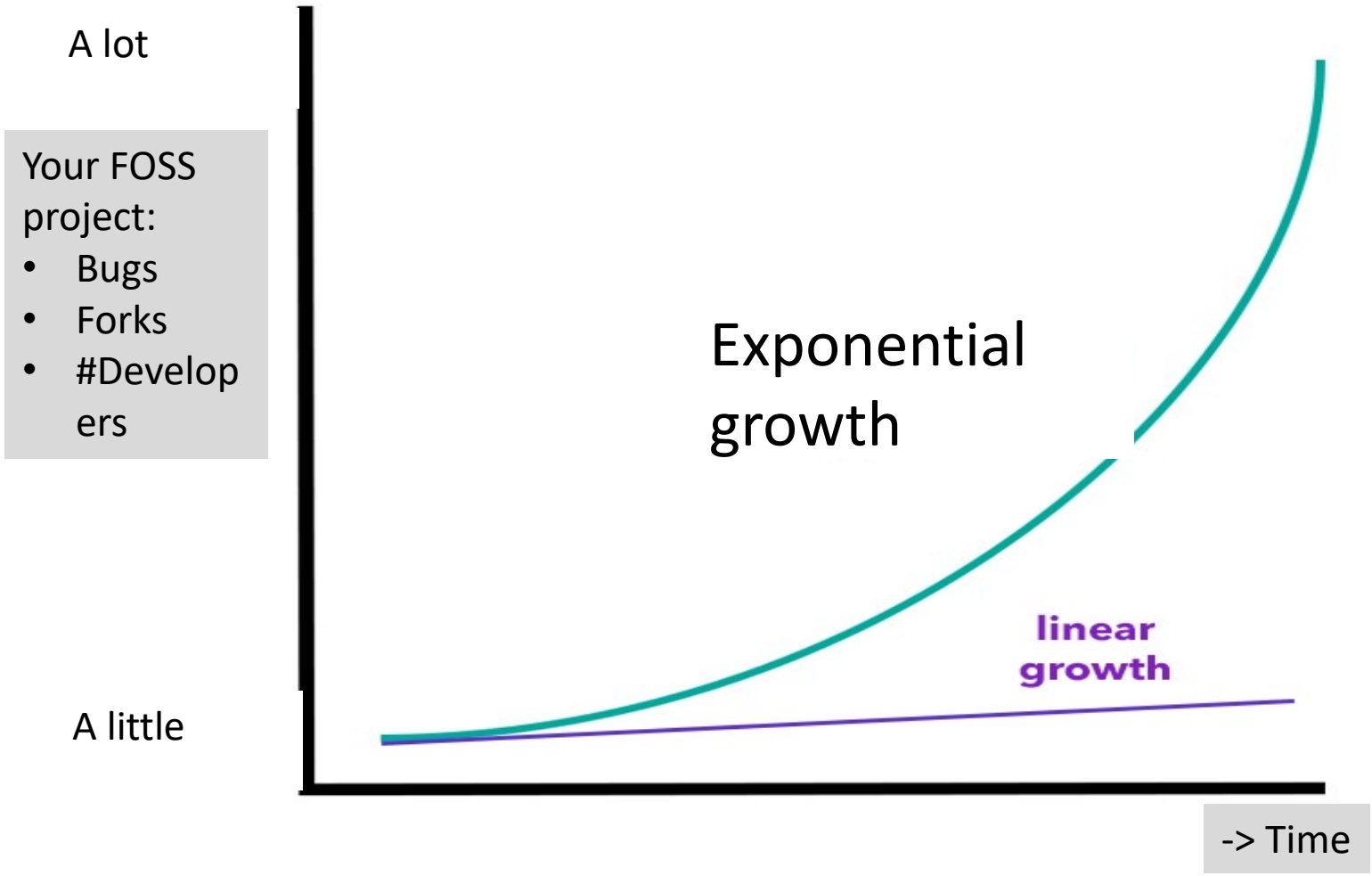
Only a few factors make problems complex and hard to solve:

- Humans (mostly behaviour)
- Time (and delays in time)

Humans

Humans

Time: Exponential behaviour



F5:Refresh

Reset your mind! Graphics are lies!

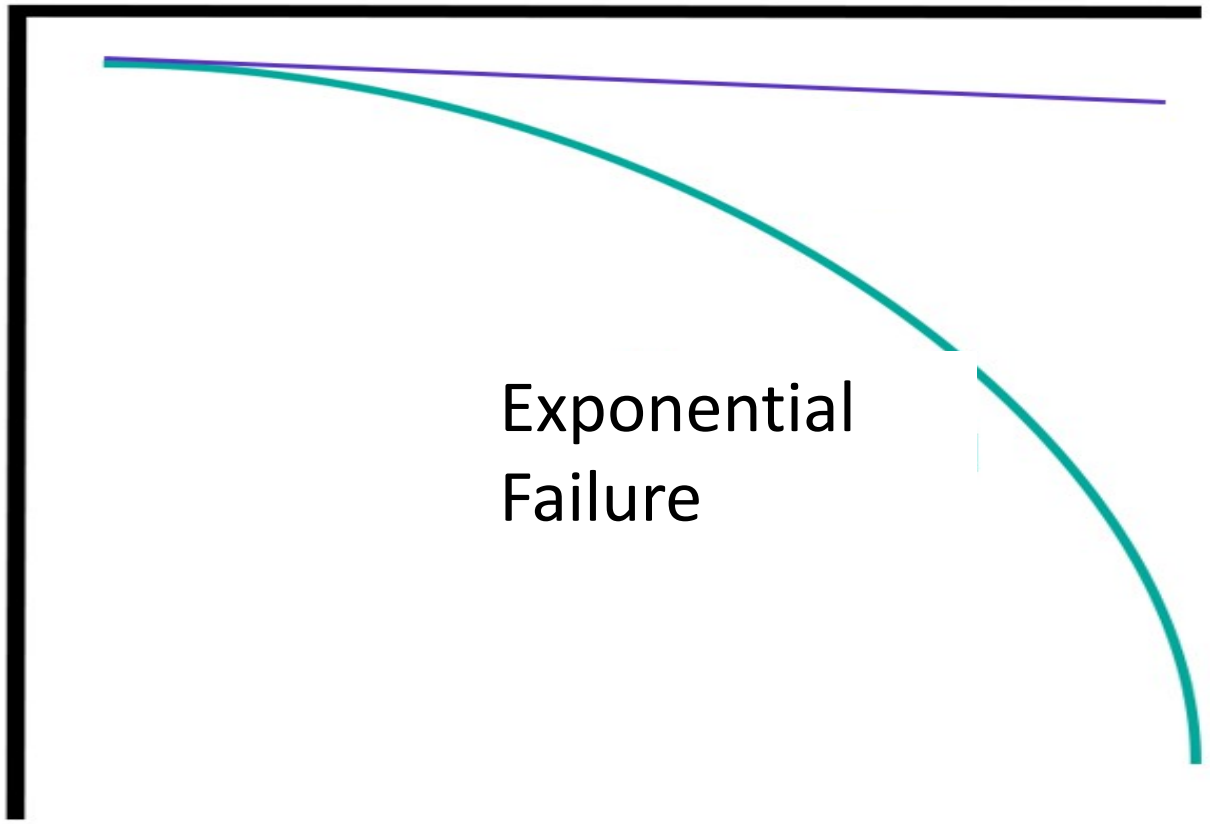
Time: Exponential (most times in reality)

-> Time

A little

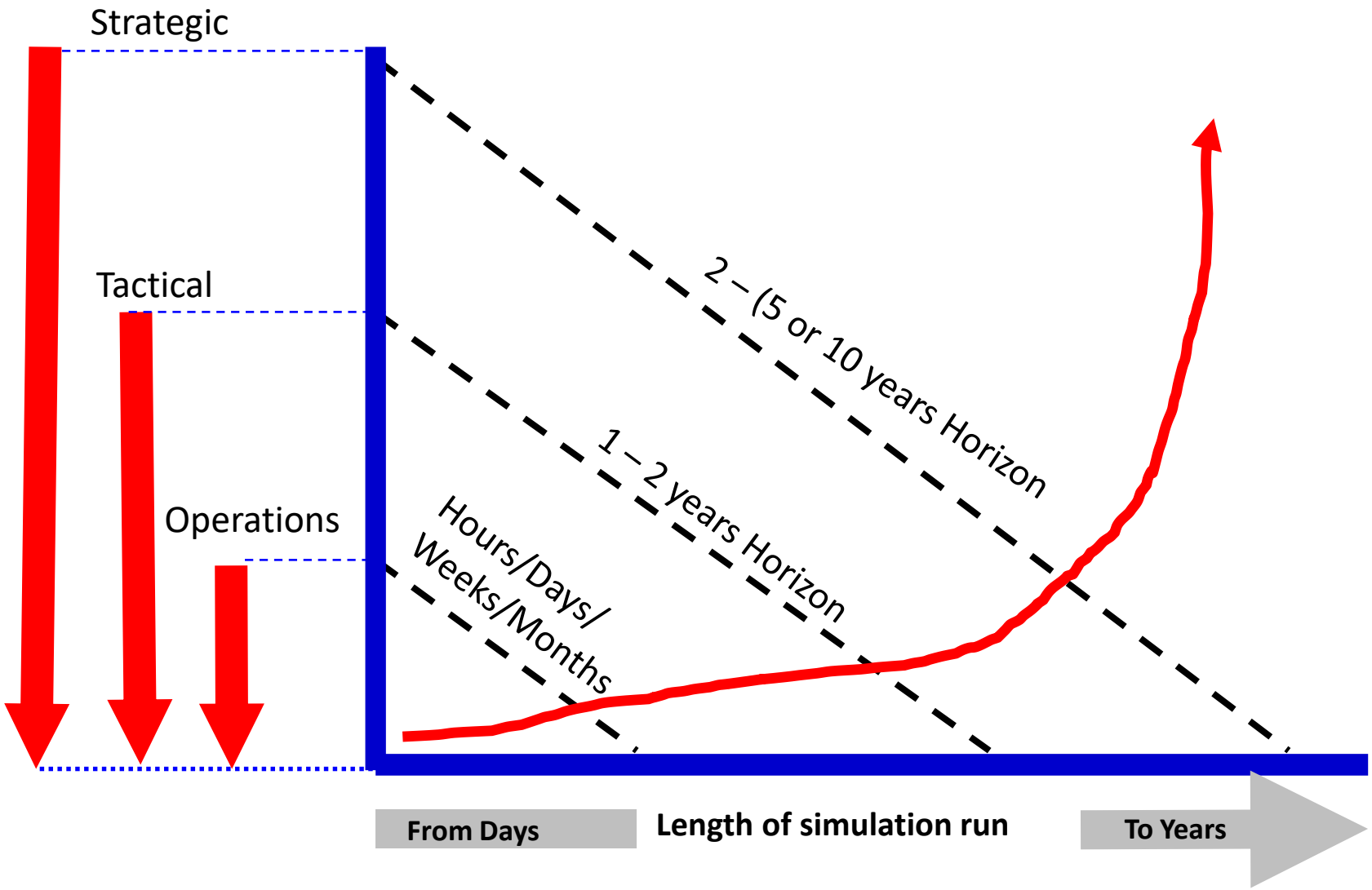
Your FOSS project:

- Bugs
- Forks
- #Developers



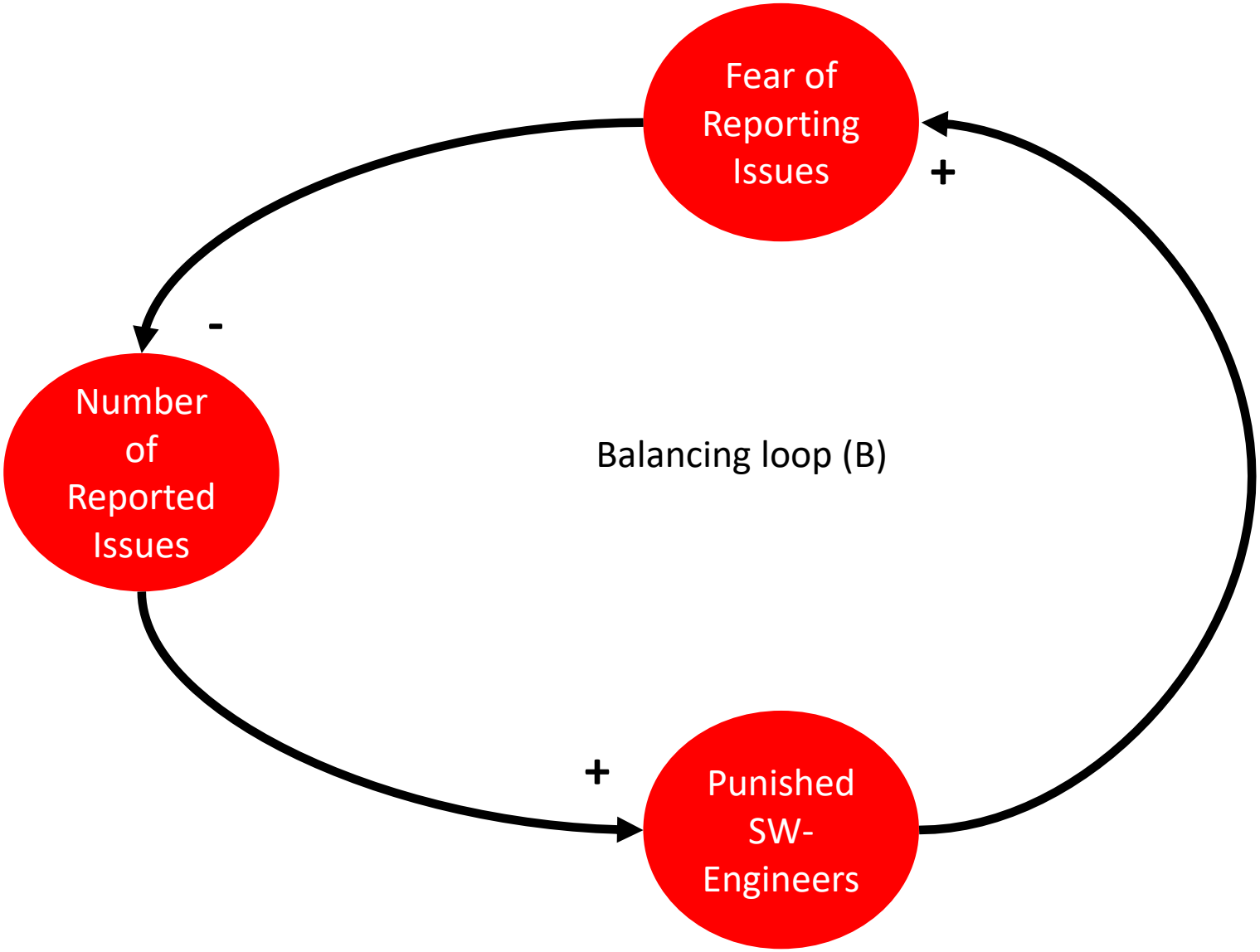
A lot

Time: Scale

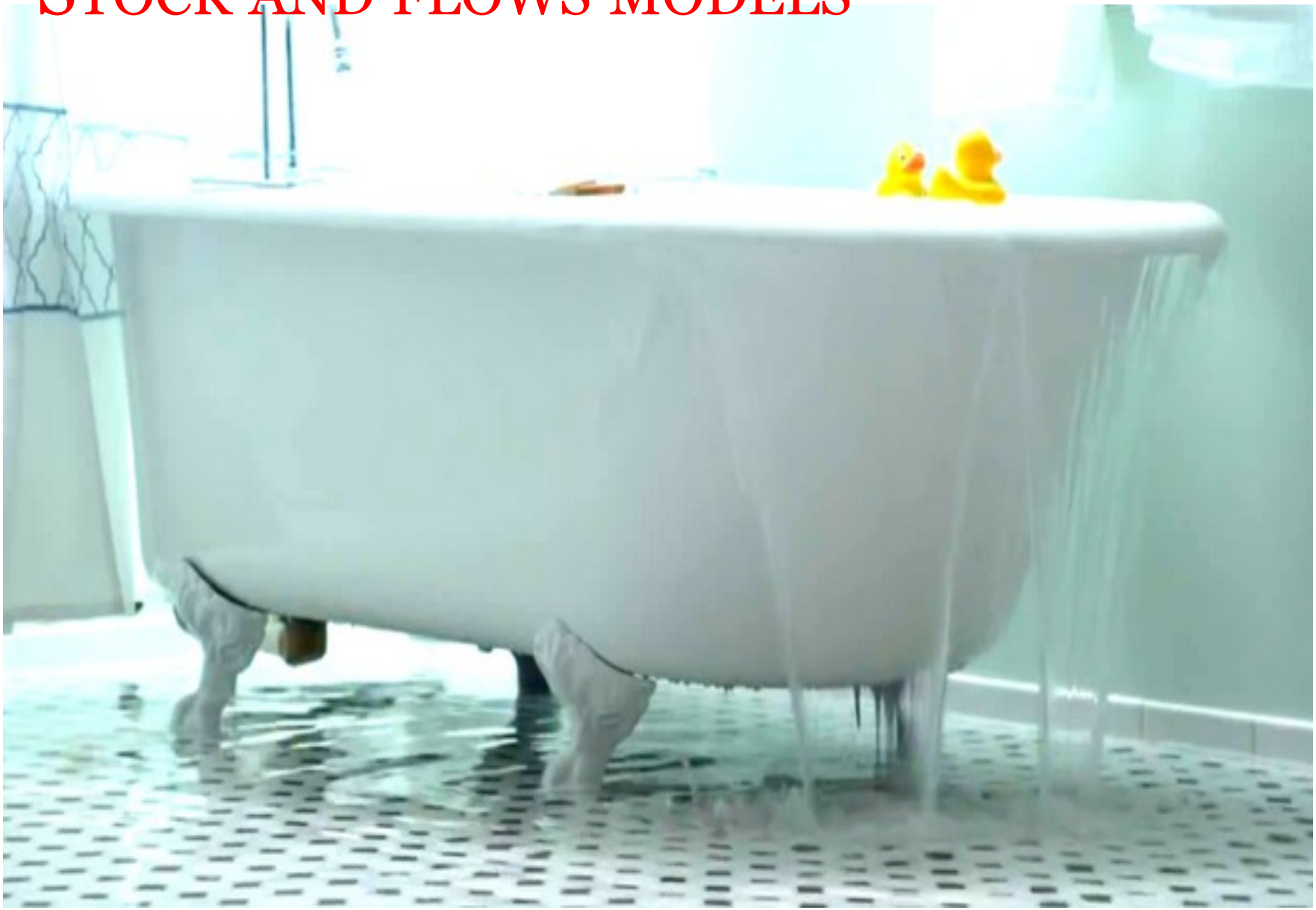

Strategic
Tactical
Operations
2 – (5 or 10 years Horizon
1 – 2 years Horizon
Hours/Days/
Weeks/Months
From Days
Length of simulation run
To Years

Core tools of System Dynamics: Sharing perceptions (mental models)

Core tools of System Dynamics:Causal Loop Diagrams

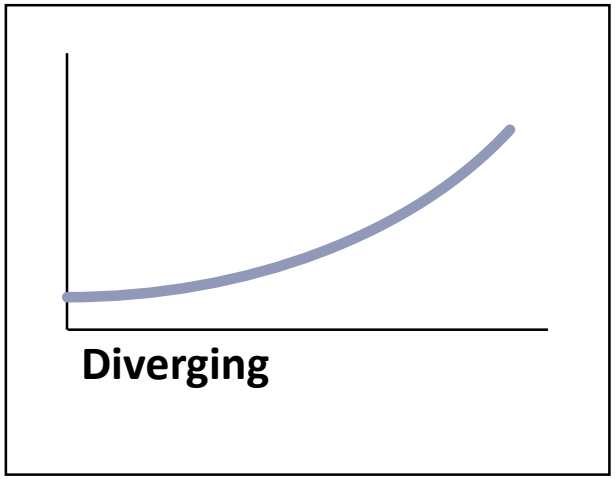


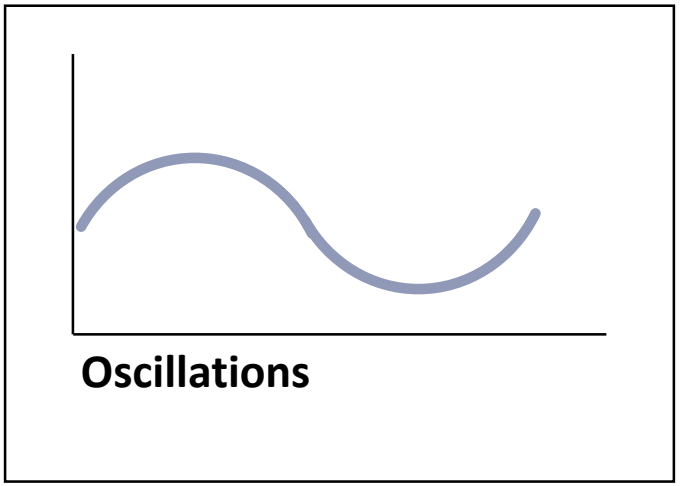
Core tools of System Dynamics: Stock and flows models

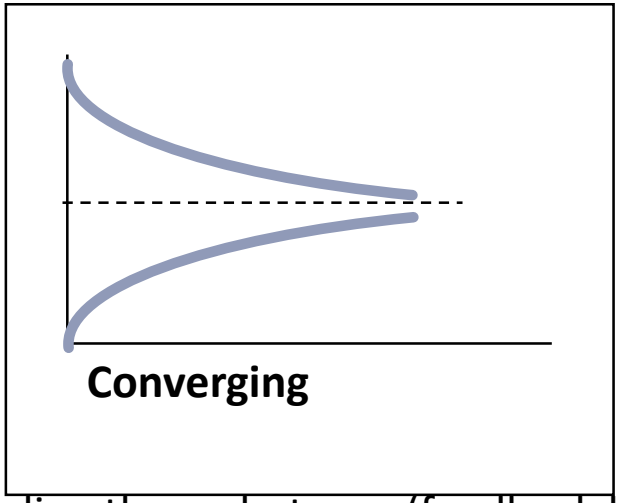


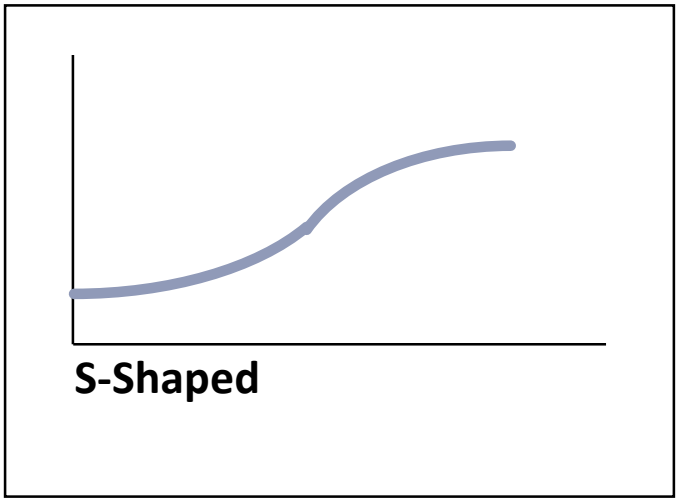
Core tools of System Dynamics Archetypes- Feedback and Behaviour

Feedback loops are linked to specific kinds of basic behaviour patterns:









Finding the archetypes (feedback loops) in non-linear systems is very hard. (=for us humans)

Core tools of System Dynamics: Simulating

Agenda

- IT projects never fail!
- What is system dynamics?
- How to model a complex IT problem? (use FOSS?)
- FOSS software for system dynamics
- Advantages and disadvantages of using FOSS tools for problem solving

How to model a complex IT problem

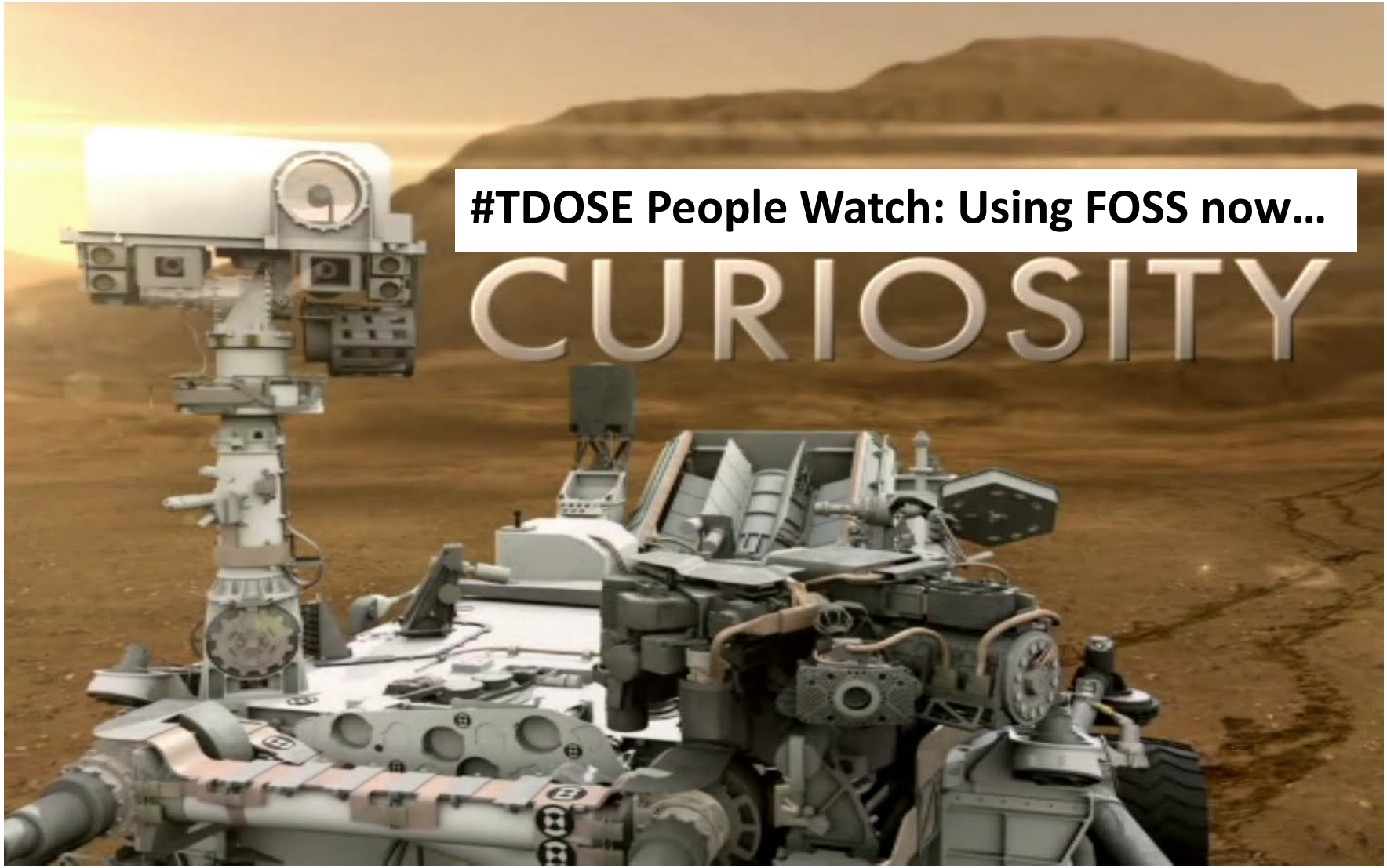
Use the Tools!

- Rich picture (share your mental model)
- Conversation
- Discussion (relationship / causality / increase or decrease)
- Draw Causal Loop Diagrams (CLD's) Diagram(s)

Optional:

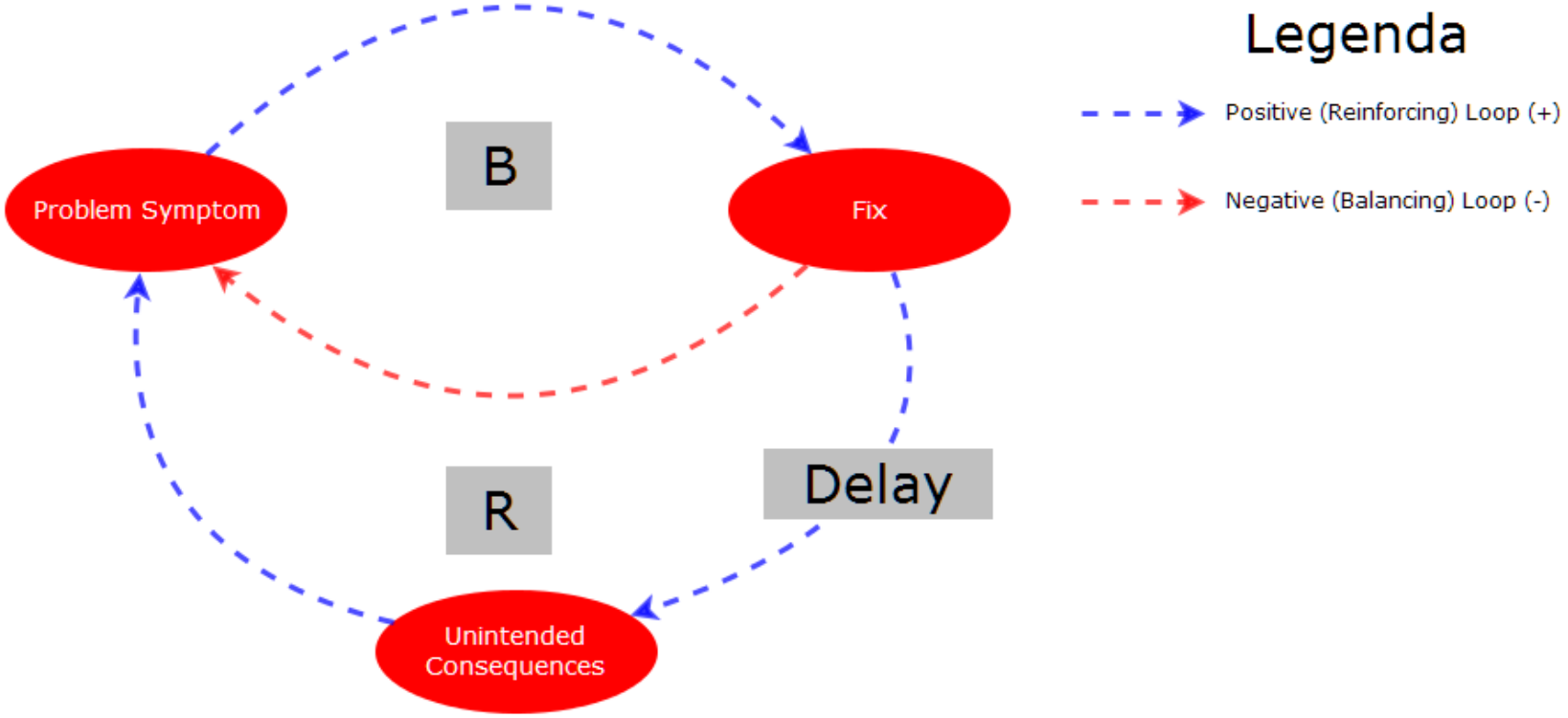
- Create Stock/Flow diagram(s)
- Calculate behaviour / outcome (use good! FOSS Software)
- Play with model

DEMO!



Simple Model

Causal Loop Diagram



Example: Increasing IT hiring to augment existing experienced staff, but then finding that the experienced staff's time is largely consumed by bringing the new hires up to speed, resulting in a sharp loss in productivity.

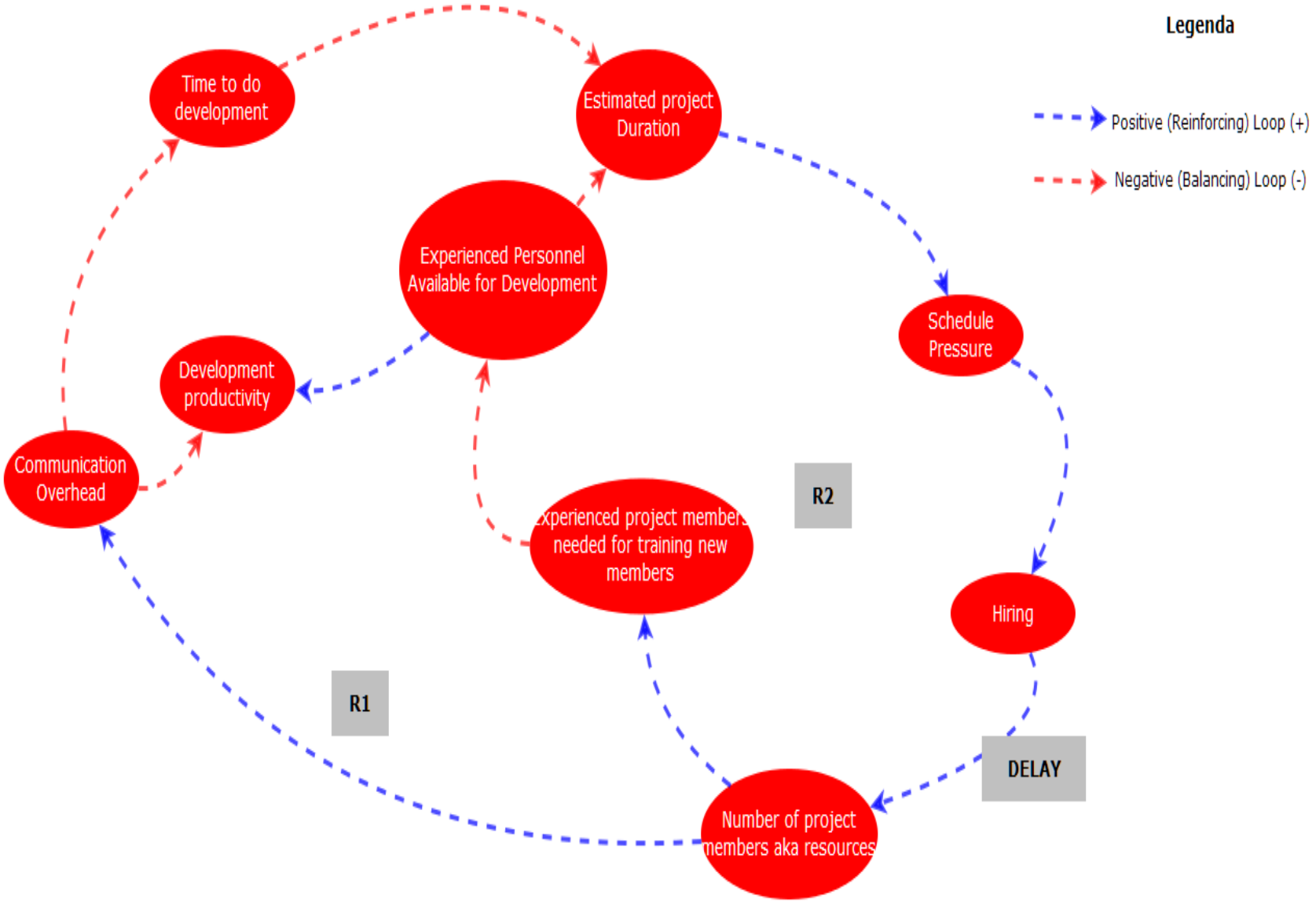
More complex model: Brooks Law

Brooks' Law: Adding manpower to a late software project makes it later

…But is it really true?

Brooks' Law is well known in the software engineering community due to the ground-breaking book, "The Mythical Man Month: Essays on Software Engineering" (https://archive.org/details/mythicalmanmonth00fred)

Brooks Law


Legenda
Positive (Reinforcing) Loop (+)
Negative (Balancing) Loop (-)
Time to do development
Estimated project Duration
Experienced Personnel Available for Development
Schedule Pressure
Development productivity
Communication Overhead
Experienced project members needed for training new members
R2
Hiring
R1
DELAY
Number of project members aka resources

Agenda

- IT projects never fail!
- What is system dynamics?
- How to model a complex IT problem?
- FOSS software for system dynamics
- Advantages and disadvantages of using FOSS tools for problem solving

FOSS AND COMPLEX PROBLEM SOLVING

Problem solving = (too much) nasty complex mathematics

Problem solving = Collaboration with other people

Problem solving = Universities/Science and FOSS

(Always hard: Technical people and non-technical people speak and think differently)

FOSS is still not the leading standard in the scientific world.

FOSS TOOLS AND COMPLEX PROBLEM SOLVING

Developing a user friendly tool for complex problem solving is hard. Great mathematics FOSS algorithms help, but is not enough.

What makes it hard?

- Math -)
- UI:User friendly
- Platform: installation == (hell | forbidden | requires time & Knowledge)

Best FOSS tools currently around for SD:

- **InsightMaker**
- **Notebooks**
- NetLogo *(Is it because I do not like Java anymore?)*

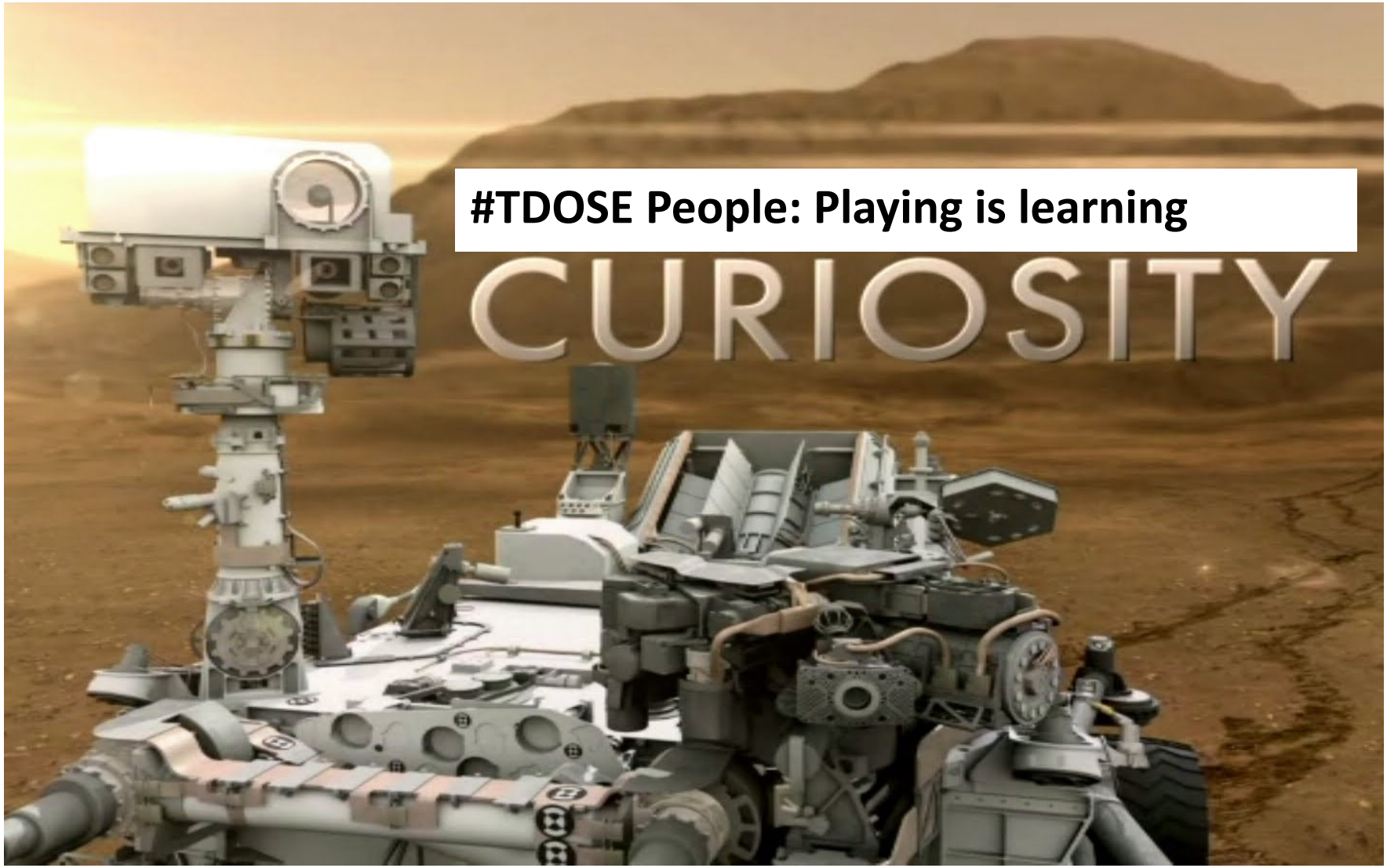
INSIGHTMAKER

A feature rich modelling suite in the browser:

- Core features:
  - System dynamics
  - Agent Based Modeling
  - Diagramming and Rich Pictures
  - And more!

- Insight maker models are web pages
- Easy sharing of work (mail, link, post on social network etc)
- FULLY browser based (all JS works in your browser)
- Clone Insight ('work') of others and improve / build upon ('the oss-way')
- Export options (CSV, models, JSON, CSS-style sheets)

DEMO!



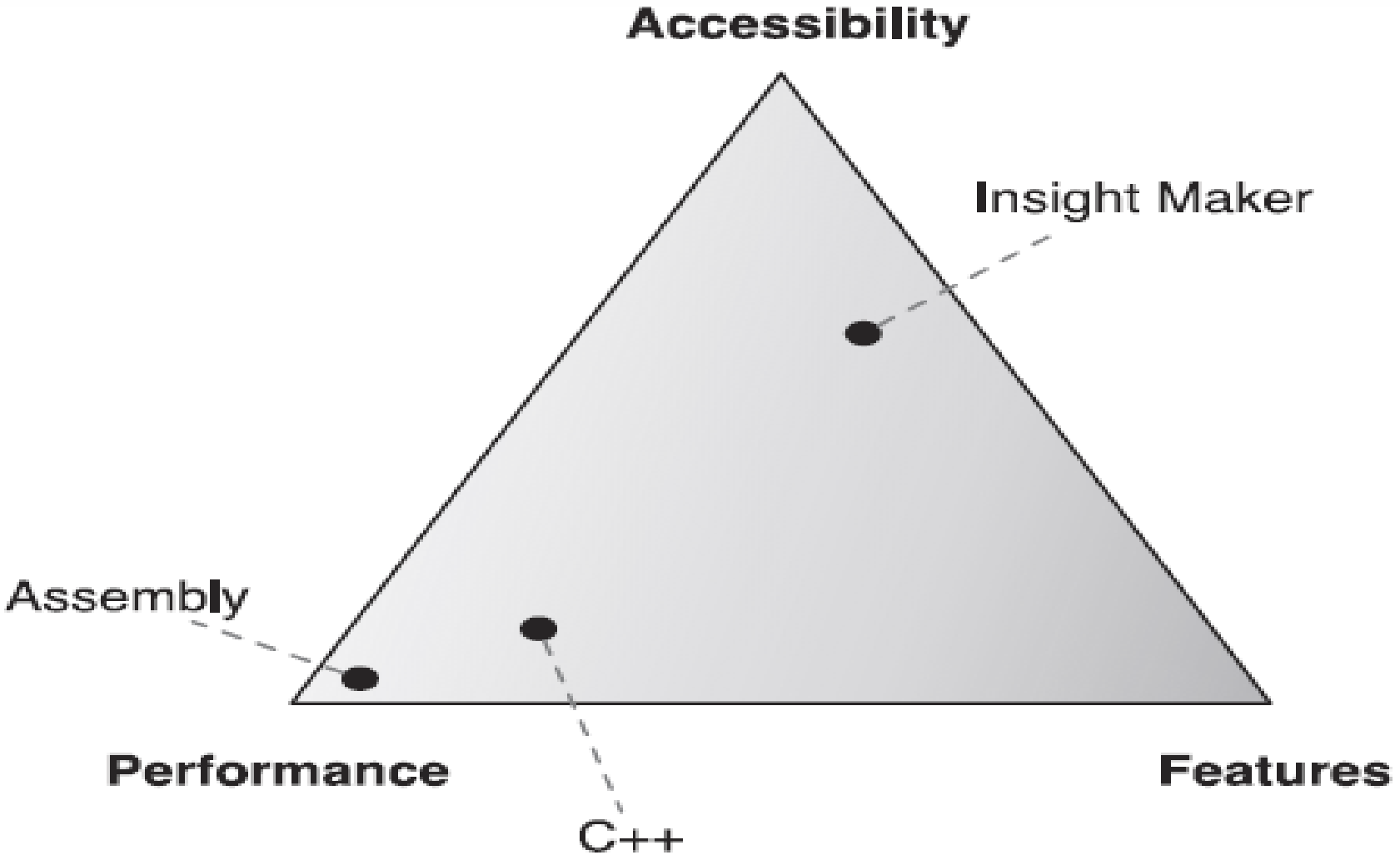
Insight Maker Technical Architecture

- Client (browser-based) / Server (minimal, only for storing)
- Server:
  - Stores models
  - Manages user accounts
  - Manages collaborative editing
- Client:
  - Model construction
  - Simulation
  - Result display
  - Client-side code runs the simulation

And since you DONOT want malware, all software is AGPL (you can run it on your own server)

Guiding principles behind Insight Maker

Trade-off between: performance, features, and accessibility

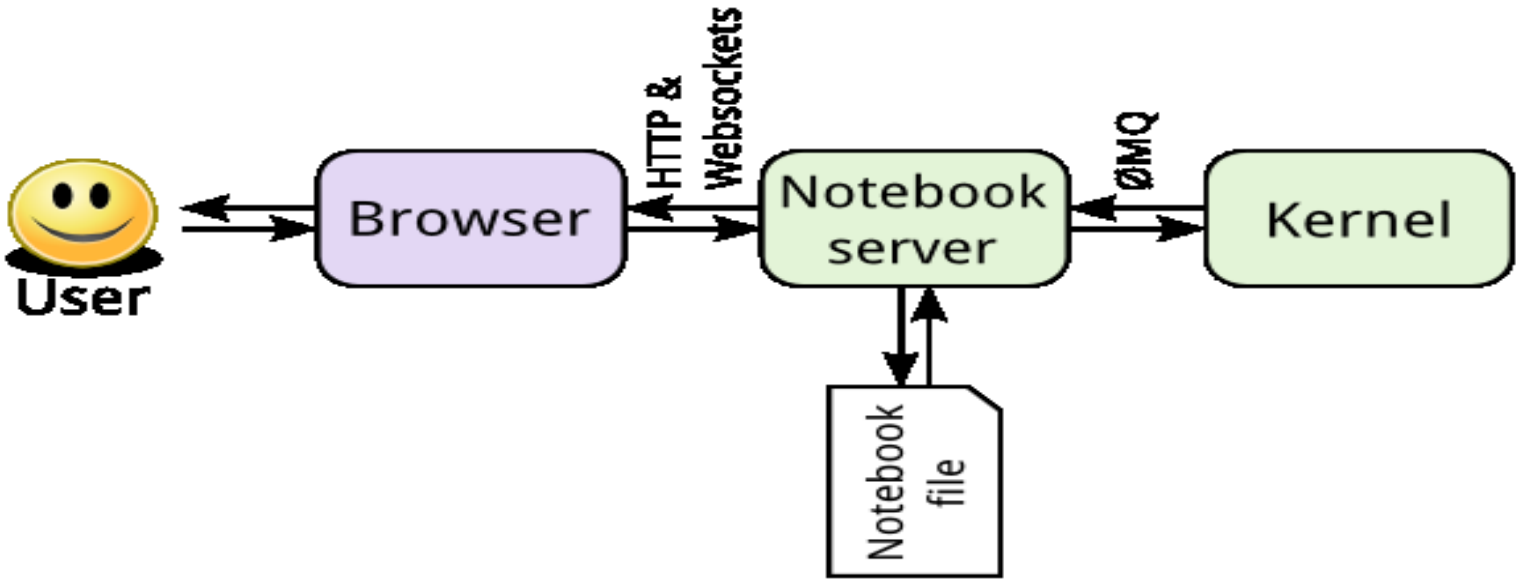


THE NOTEBOOK

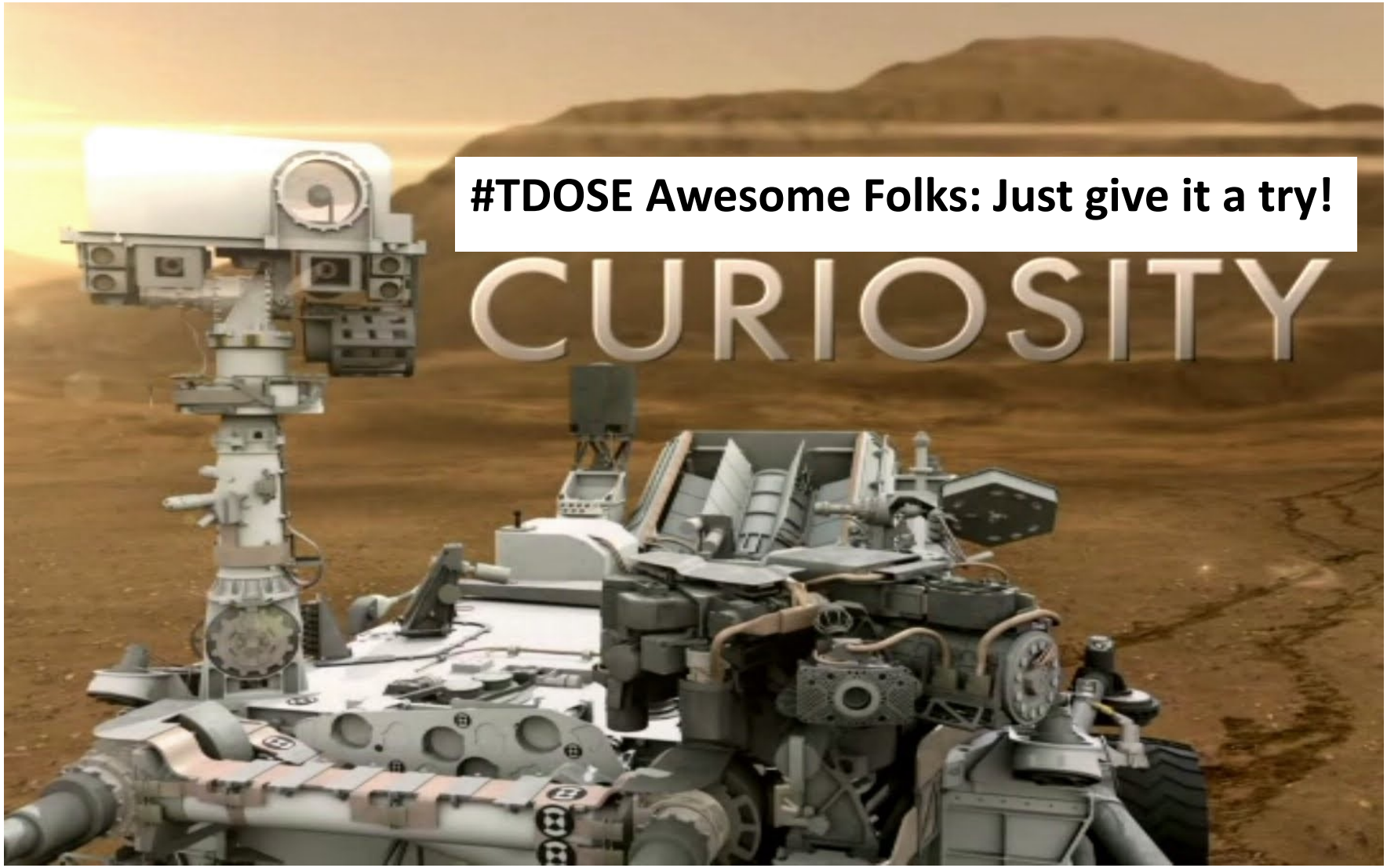
The notebook

The Jupyter Notebook

- Successor of IPython Project in 2014
- A web application to create and share documents that contain live code, equations, visualizations and explanatory text.
- Notebook has support for over 40 programming languages
- Reproducible science, scientific computing, data science ('R'), data visualization and more



DEMO!



Agenda

- IT projects never fail!
- What is system dynamics?
- How to model a complex IT problem?
- FOSS software for system dynamics
- Advantages and disadvantages FOSS tools for problem solving (and recap)

Advantages and disadvantages of FOSS for Complex IT problems

What is your experience / opinion?

FUTURE?!

- Standard for on input/output of models (Shall InsightMaker (=we) set the new standard, so that commercial vendors will follow?)
- More people working with AND on InsightMaker? (Maintenance !!)
- More notebooks online (cc-licensed)! (already 500k on github)
- JupyterLab (next-gen Jupyter notebooks)
- Easier integration and exchange of tools and results
- Easy-to-use FOSS tools for Data Analyse ('big-data') and merge of data analytics with system dynamic modeling.

Summary

The real power understanding complex IT problems by system dynamics is only possible through:

- Simulation

Be aware: All models are WRONG (John D. Sterman - SD guru @MIT)

Summary

System Dynamics (SD) is a method to manage complex systems characterized by:

- nonlinear dynamics
- feedback
- time delays
- soft factors
- interdisciplinary aspects

Using FOSS makes SD easier to use!

(And can make it easier to really solve complex IT problems together)

Disclaimer: Not all type of problems are suitable to approach with system dynamics!

THANK YOU!

More information?

Call me : +31 [0] 6 22869536 of
Mail : info@organisatieontwerp.nl
Twitter : @maikelmardjan

(Sometimes available for solving your *real nasty* complex IT problems!)