# Technology Assurances and Sandboxes


@joshuaellul

linkedin.com/in/joshuaellul

Dr **Joshua Ellul**

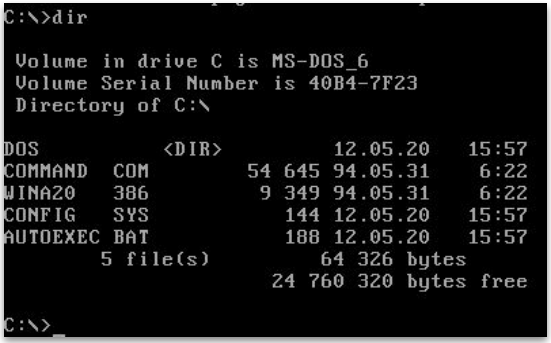
MDIA Malta Digital Innovation Authority — **Chairman**

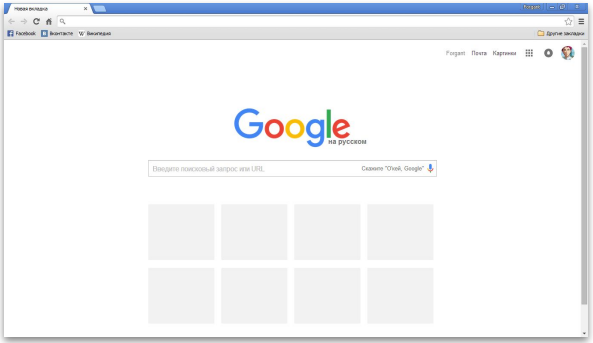
L-Università ta' Malta Centre for Distributed Ledger Technologies — **Director**

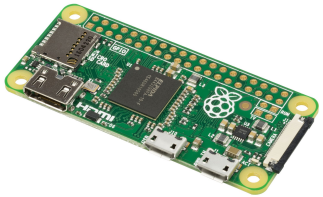

MFSA Malta Financial Services Authority

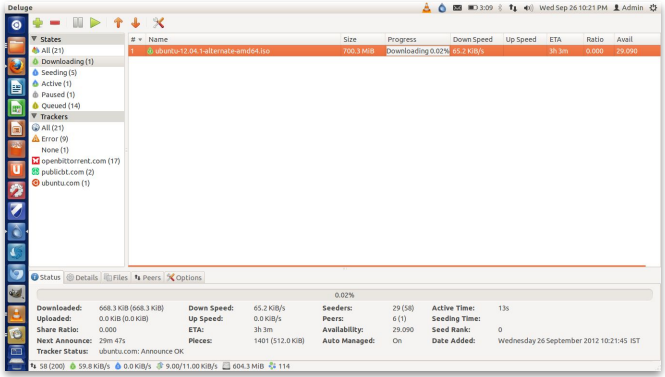
FinanceMalta Effective | Secure | Skilled

# It’s Software



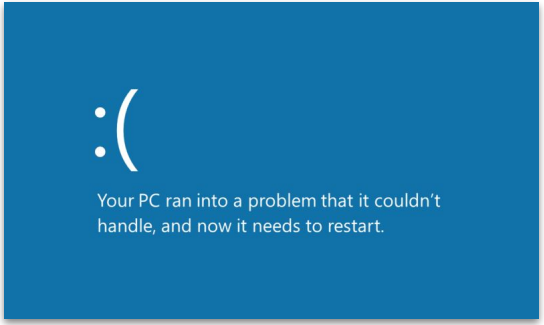



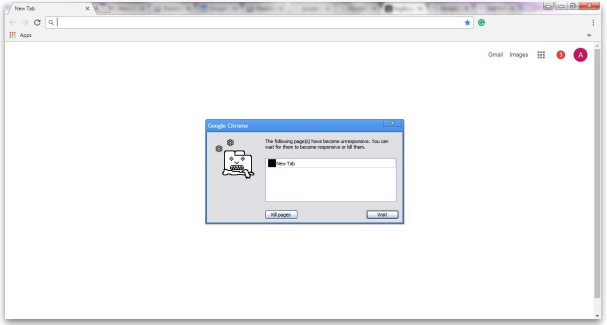



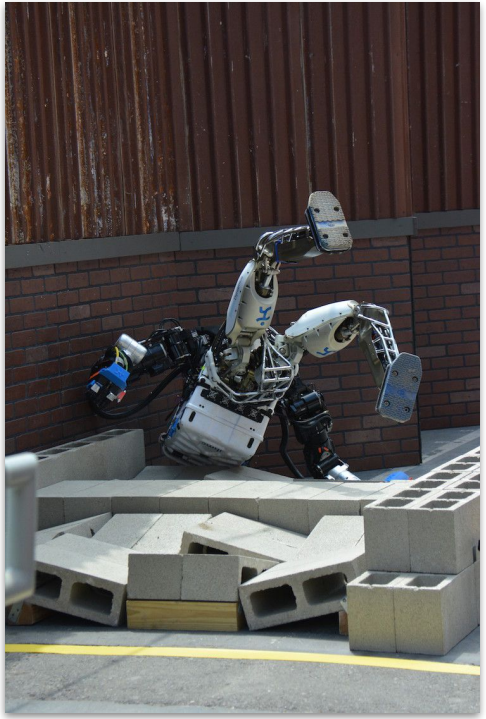


https://en.wikipedia.org/wiki/Mac_OS_X_Leopard#/media/File:Leopard_Desktop.png
https://upload.wikimedia.org/wikipedia/commons/a/a1/SoftBank_pepper.JPG
https://upload.wikimedia.org/wikipedia/commons/5/50/Google_Chrome_45.0.2454.85_Windows_XP.PNG
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRl0J-9nMu_enEsX8BraYYQre4rHVkqUVKGvbwyMQVh3TP5rBR6EQ
https://upload.wikimedia.org/wikipedia/commons/0/07/MS-Dos_screenshot.png
https://upload.wikimedia.org/wikipedia/commons/b/b1/Deluge_1.3.5_Ubuntu_12.04_LTS.png
https://upload.wikimedia.org/wikipedia/commons/7/7e/Raspberry-Pi-Zero-FL.jpg

# Software just works, right?





https://upload.wikimedia.org/wikipedia/commons/thumb/5/57/Blue_Screen_of_Death.png/800px-Blue_Screen_of_Death.png
https://upload.wikimedia.org/wikipedia/commons/7/7e/OS_X_10.11_Beta_Beach_Ball.jpg
https://techysupport.co/wp-content/uploads/2018/04/google-chrome-not-working.png
https://upload.wikimedia.org/wikipedia/commons/a/a6/Kernel_panic_Android_Linux.jpg
https://laughingsquid.com/wp-content/uploads/2015/06/IHMC-Crossing-Rubble-Day-1-Photo-31.jpg

# Sometimes, it’s not just about software

- Logic often in Hardware and Firmware
  - Even for hardware is very often implemented as code
  - Hardware cannot be updated (without replacing)
  - Firmware often inaccessible

- Hardware (and Firmware) **can contain bugs** too!
  - Pentium **FDIV** bug
    - On affected processors may result in incorrect division operations

https://en.wikipedia.org/wiki/Pentium_FDIV_bug

# Often used for Critical Systems

- Systems when not executed correctly, may:
  - lose or leak sensitive data
  - jeopardise whole operations
  - result in large financial losses
  - lead to **loss of life**, or **material damage**

# Infamy

- NASA’s Spirit rover became unresponsive a few weeks after landing on Mars because it had stored too many files
- The Therac-25 radiation therapy machine was responsible for killing at least 5 patients, due to administering massive overdoses of radiation
- Y2K
  - Year 2038 (32 bit signed integer since 1st January 1970)
  - Year 2106 (32 bit unsigned integer since 1st January 1970)
    - Will affect how Bitcoin stores block time (for current implementation)
- …

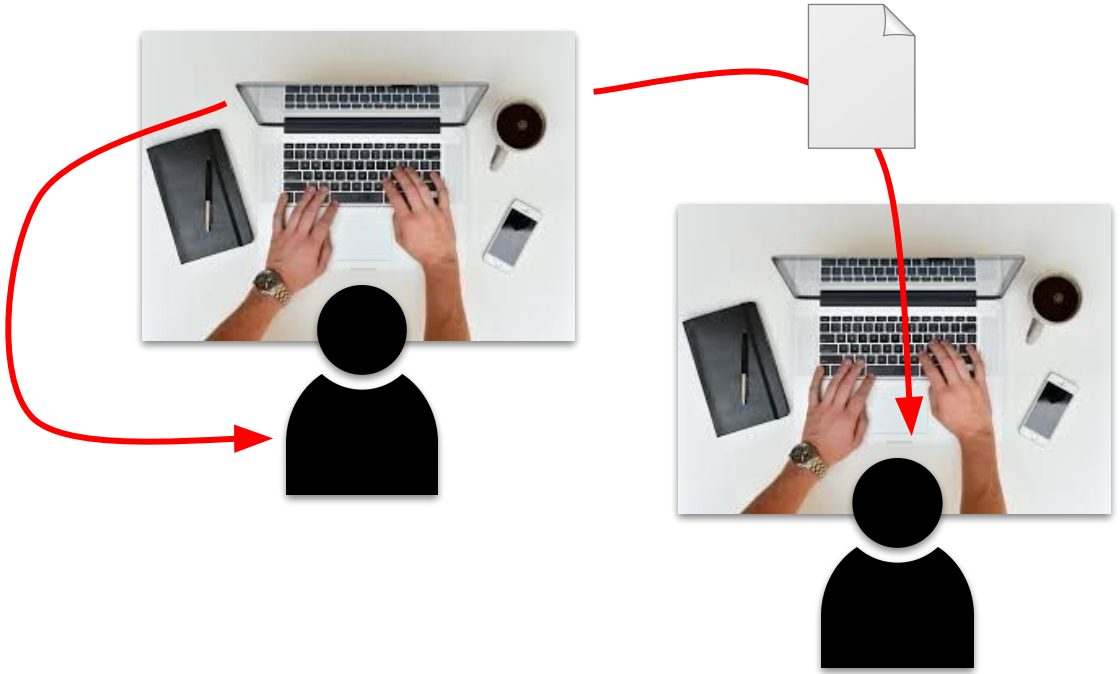
https://en.wikipedia.org/wiki/List_of_software_bugs
https://en.bitcoin.it/wiki/Block_timestamp

# Assurances

- **Developer Support Tools**

# Assurances

- Developer Support Tools
- **Testing**



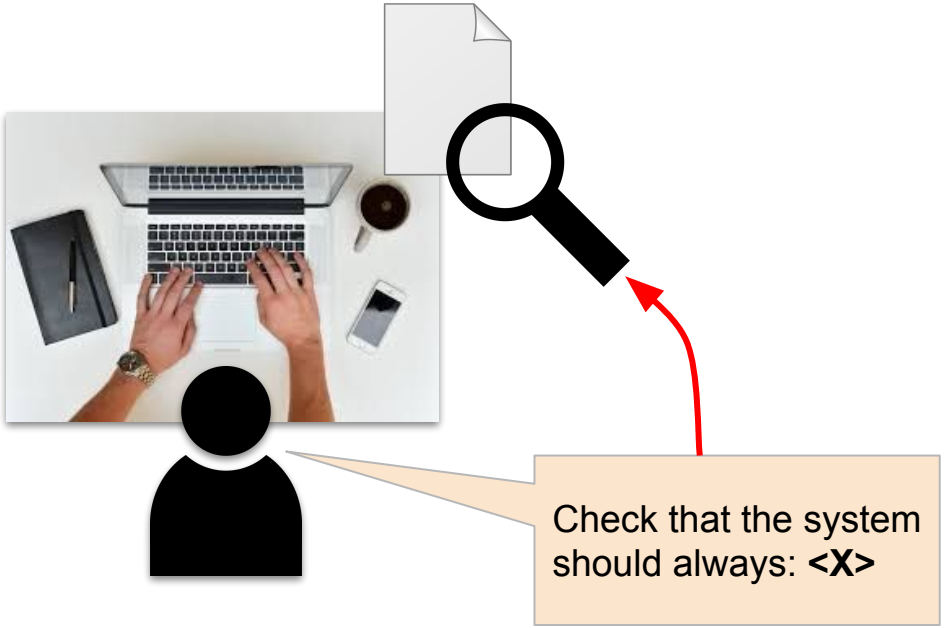
https://images.pexels.com/photos/5199/hands-coffee-cup-apple.jpg?auto=compress&cs=tinysrgb&dpr=2&h=750&w=1260
https://cdn.pixabay.com/photo/2016/02/17/16/32/person-1205346_960_720.png
http://res.publicdomainfiles.com/pdf_view/70/13930287014093.png
https://upload.wikimedia.org/wikipedia/commons/thumb/b/b7/PICOL_icon_Search.svg/1024px-PICOL_icon_Search.svg.png
https://upload.wikimedia.org/wikipedia/commons/thumb/b/bd/Checkmark_green.svg/1180px-Checkmark_green.svg.png
https://cdn.pixabay.com/photo/2016/03/31/21/59/bug-1296767_960_720.png

# Testing - as good as the coverage

- Developer Support Tools
- **Testing**

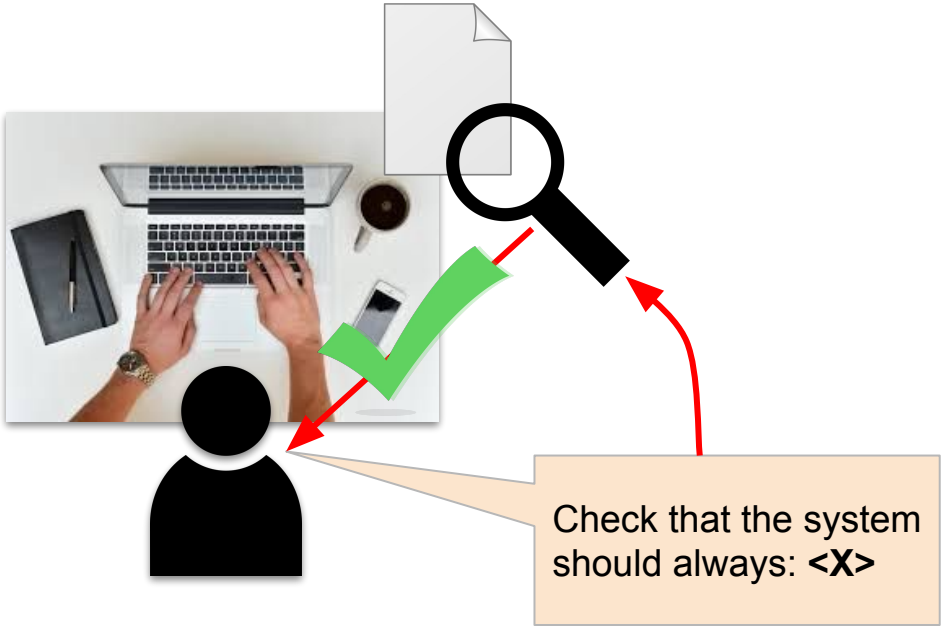
# Software Assurances

- Developer Support Tools
- Testing
- **Static verification**



# Software Assurances

- Developer Support Tools
- Testing
- **Static verification**

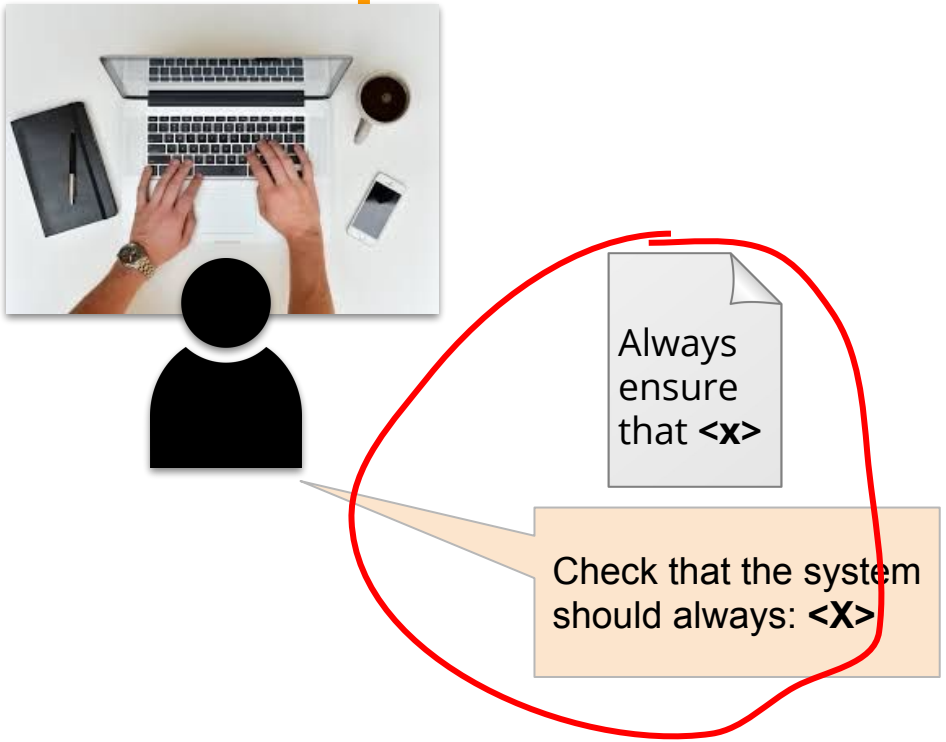


# Software Assurances

- Developer Support Tools
- Testing
- Static verification
- **Runtime verification**

# Verification - as good as the specification

- Developer Support Tools
- Testing
- **Static verification**
- **Runtime verification**



# Software Assurances

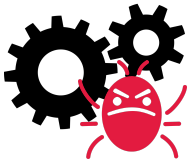
- Developer Support Tools
- Testing
- Static verification
- Runtime verification
- **Code fixes** (where possible)

# Software Assurances

- Developer Support Tools
- Testing
- Static verification
- Runtime verification
- **Code fixes** (where possible)

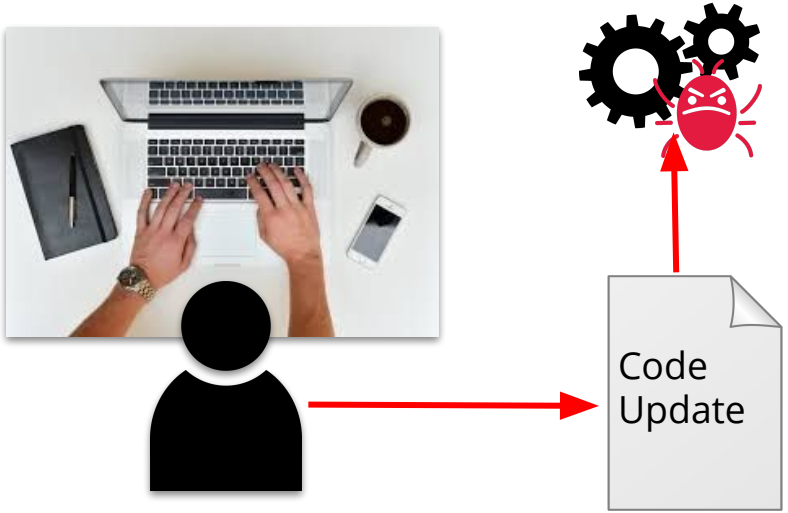
# Software Assurances

- Developer Support Tools
- Testing
- Static verification
- Runtime verification
- **Code fixes** (where possible)



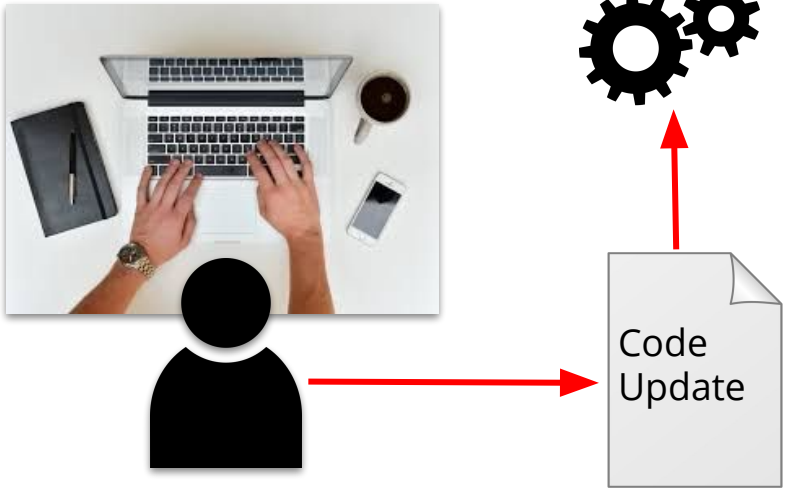
# Software Assurances

- Developer Support Tools
- Testing
- Static verification
- Runtime verification
- **Code fixes** (where possible)



# Software Assurances

- Developer Support Tools
- Testing
- Static verification
- Runtime verification
- **Code fixes** (where possible)




https://images.pexels.com/photos/5199/hands-coffee-cup-apple.jpg?auto=compress&cs=tinysrgb&dpr=2&h=750&w=1260
https://cdn.pixabay.com/photo/2016/02/17/16/32/person-1205346_960_720.png
http://res.publicdomainfiles.com/pdf_view/70/13930287014093.png
https://upload.wikimedia.org/wikipedia/commons/thumb/b/b7/PICOL_icon_Search.svg/1024px-PICOL_icon_Search.svg.png
https://upload.wikimedia.org/wikipedia/commons/thumb/b/bd/Checkmark_green.svg/1180px-Checkmark_green.svg.png
https://cdn.pixabay.com/photo/2016/03/31/21/59/bug-1296767_960_720.png

# Software Assurances

- Developer Support Tools
- Testing
- Static verification
- Runtime verification
- Code fixes (where possible)

- Is this good enough?
  - For many applications, yes

- But **not**, **for certain applications**:
  - **Safety critical**
  - When code **fixes** are **not possible**

# Smart Contracts and User Assurances

- Decentralised guaranteed execution of logic
  - Code ‘cannot’ be changed → Immutable
  - Parties ‘know’ what they are agreeing to
- Code is available for ‘all’ to see

- Smart contract **logic is often simple**

# Even Simple Code Could be Buggy

- Smart contracts can be buggy
  - What do we usually do?
    - Developer Support Tools ✓
    - Testing ✓
    - Static verification ✓
    - Runtime verification ✓
    - Code fixes, **remember**:
      - Code ‘cannot’ be changed → Immutable
      - Parties ‘know’ what they are agreeing to
      - **So if there’s a bug, is it there forever?**

# Innovative Solutions brings ‘Unknowns’

- Innovation -  new, unexplored territories: brings unknowns
- How can we foster innovation, yet keep levels of assurances?
    - We need a safe place to learn

# Sandbox

- The good ol' days: a box that contains sand for children to play in
- Software development:

A **sandbox** is a testing environment that isolates untested code changes and outright experimentation from the production environment or repository,[1] i

- Regulatory sandbox:

Recently a number of jurisdictions have established Regulatory Sandboxes - environments which support and facilitate innovation whilst safeguarding consumer protection, market integrity and financial soundness.

What do they have in common?

If we define Technological sandboxes, how can we be sure that the technology stays within its parameters?

# Similar to Hardware and Critical Systems?

- (Often) Cannot update
- Bugs could have critically high costs

- What is typically done in similar industries?
  - **Independent audits** to raise levels of assurances

# Want to stay in touch?

Potential to collaborate
Talks
Pilot projects
Research proposals and grants
Training courses

MDIA Malta Digital Innovation Authority
joshua.ellul@mdia.gov.mt

L-Università ta' Malta
Centre for Distributed Ledger Technologies
joshua.ellul@um.edu.mt

@joshuaellul

linkedin.com/in/joshuaellul