Architektury softwarových systémů
Architecture of Software Systems

Martin Rehák, David Šišlák
Martin Grill, Ján Jusko, Jan Stiborek, Martin Vejmelka
Agent Technology Center
Department Of Computer Science and Engineering

Teachers

- Lectures
  - Martin Rehak (rehak@agents.fel.cvut.cz)
  - David Sislak (sislak@agents.fel.cvut.cz)
- Labs
  - Martin Grill (Lead)
  - Jan Jusko
  - Jan Stiborek
  - Martin Vejmelka

Topics – Lectures

14.2. [MR] Uvod: Architektury softwarových systému, komponentové a distribuované architektury

21.2. [DS] Koncepce jazyku na bázi virtualního stroje, srovnáni s jinými jazyky, přehled výhod a nevýhod; kompilace, decompilery, obfuscatory, classloaders, reflektivní operace

28.2. [DS] Selected design patterns

06.3. [MR] Design patterns for distributed systems

13.3. [DS] Vlákna, synchronizace, atomické typy, non-blocking algoritmy

20.3. [MR] RMI - architektura, podpurné komponenty, vzdálená invokace, komunikace mezi procesy

27.3. [MR] Komponentové modely, Distribuované komponenty, CORBA

03.4. [MR] Vyhledávání služeb, dynamická kompozice, Redundance, design spolehlivých systémů

10.4. [DS] Streamy, vstupni/výstupní operace, sítová komunikace, serializace, externalizace

17.4. [DS] Datové struktury - primitiva, pole; memory management s garbage collectorem

24.4. [MR] Webové služby, service-oriented architectures

01.5. Statni svatek

10.5. [MR] Architektury pro service oriented architectures

15.5. [DS] Asynchronní architektury, producer-consumer model (messaging passing) + Agentní a multiagentní systémy

... Tento plán se bude dynamicky měnit...

Evaluation

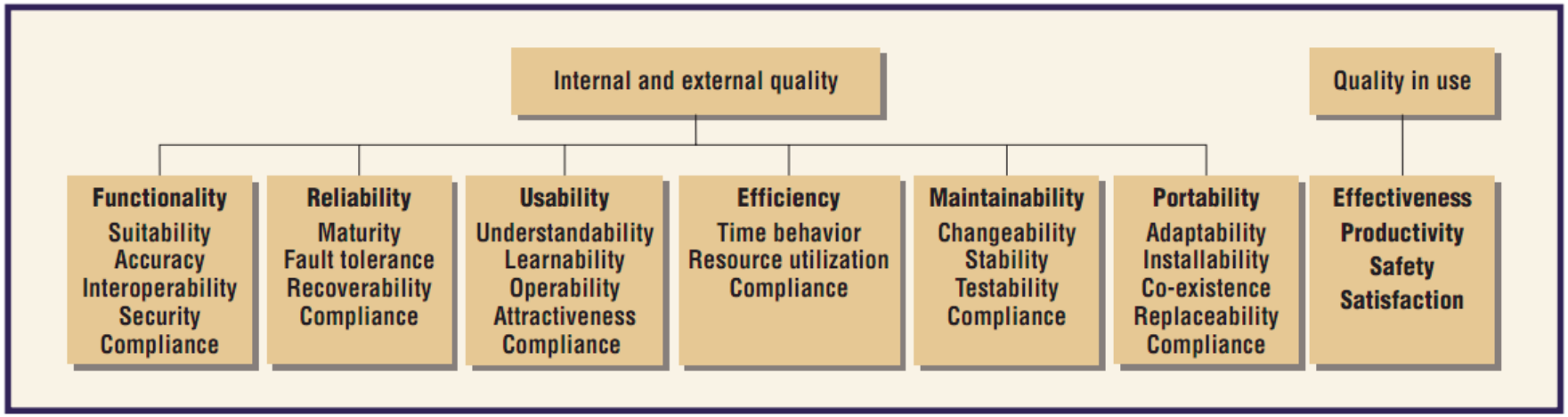
- Lab participation mandatory...as required by CTU...checked when appropriate
- Lab teachers **will not** repeat the lecture content
- Lab results are **30**% of the final grade
- Final Exam contributes another 70%
- Strict no cheating policy. Cheaters will be **failed** and **reported**.

From craftsmanship…

… to system engineering

Non-Functional Aspects of Systems

- ISO/IEC 9126 specifies the non-functional requirements on software systems
- Refined/extended in the ISO 250nn series



Jørgen Bøeg, **A New Standard for Quality Requirements, IEEE Software 2008**

ISO/IEC 9126 Categories

- Functionality
  - *existence of a set of functions and their specified properties*
- Reliability
  - *capability of software to maintain its level of performance under stated conditions for a stated period of time*
- Usability
  - *effort needed for use*
- Efficiency
  - *relationship between the level of performance of the software and the amount of resources used, under stated conditions.*
- Maintainability
  - *effort needed to make specified modifications*
- Portability
  - *transferred from one environment to another*

quoted from ISO/IEC 9126

ISO/IEC 9126 Categories

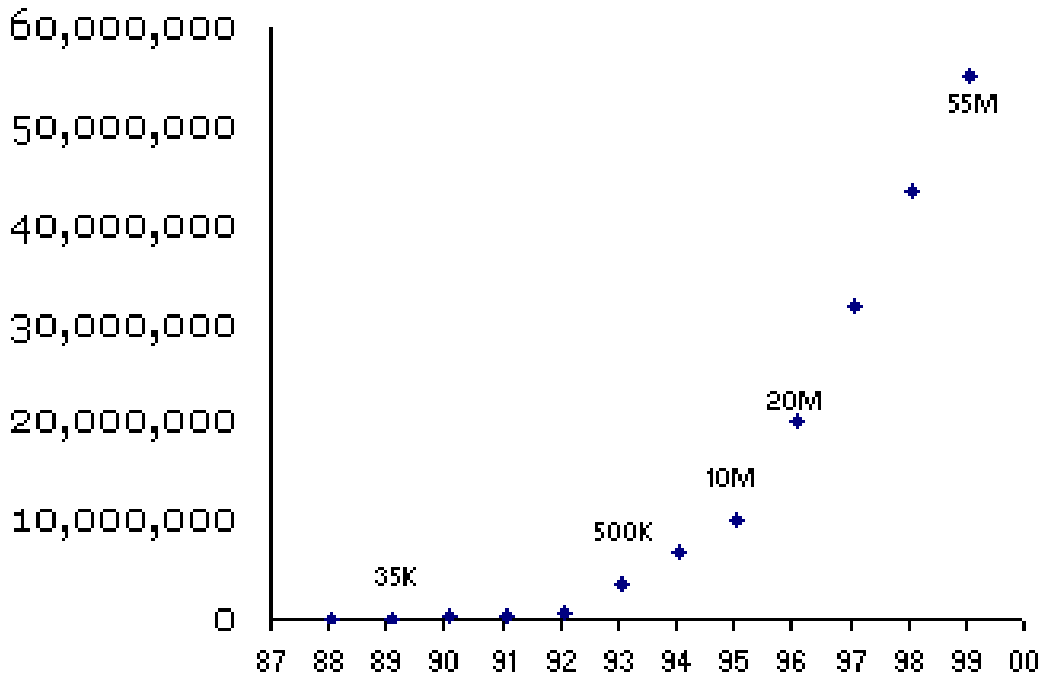
- Functionality
  - *existence of a set of functions and their specified properties*
- **Reliability**
  - *capability of software to **maintain** its level of performance under stated conditions for a stated period of time*
- Usability
  - *effort needed for use*
- **Efficiency**
  - *relationship between the **level of performance** of the software and the amount of **resources** used, under stated conditions.*
- **Maintainability**
  - *effort needed to **make specified modifications***
- **Portability**
  - *transferred from one **environment** to another*

quoted from ISO/IEC 9126

Reliability

- **Maturity**
  - *…**frequency of failure** of the software.*
- **Fault Tolerance**
  - *The ability of software to withstand (and **recover**) from component, or environmental, failure.*
- **Recoverability**
  - *Ability to bring back a failed system to **full operation**, including data and network connections.*
- Reliability Compliance
  - *"…adheres to standards, conventions, or regulations"*

Good softwre takes 10 years



Lotus Notes market penetration chart, *from Joel Spolsky, joelonsoftware.com*

Reliability

- **Maturity**
  - *…**frequency of failure** of the software.*
- **Fault Tolerance**
  - *The ability of software to withstand (and **recover**) from component, or environmental, failure.*
- **Recoverability**
  - *Ability to bring back a failed system to **full operation**, including data and network connections.*
- Reliability Compliance
  - *"…adheres to standards, conventions, or regulations"*

Efficiency

- **Time Behavior**
  - Characterizes **response times** for a given **thru put**, i.e. transaction rate.
- **Resource Utilization**
  - Characterizes **resources used**, i.e. memory, cpu, disk and network usage.
- Efficiency Compliance
- **Scalability**
  - *Relationship between Time Behavior and Resource Utilization*

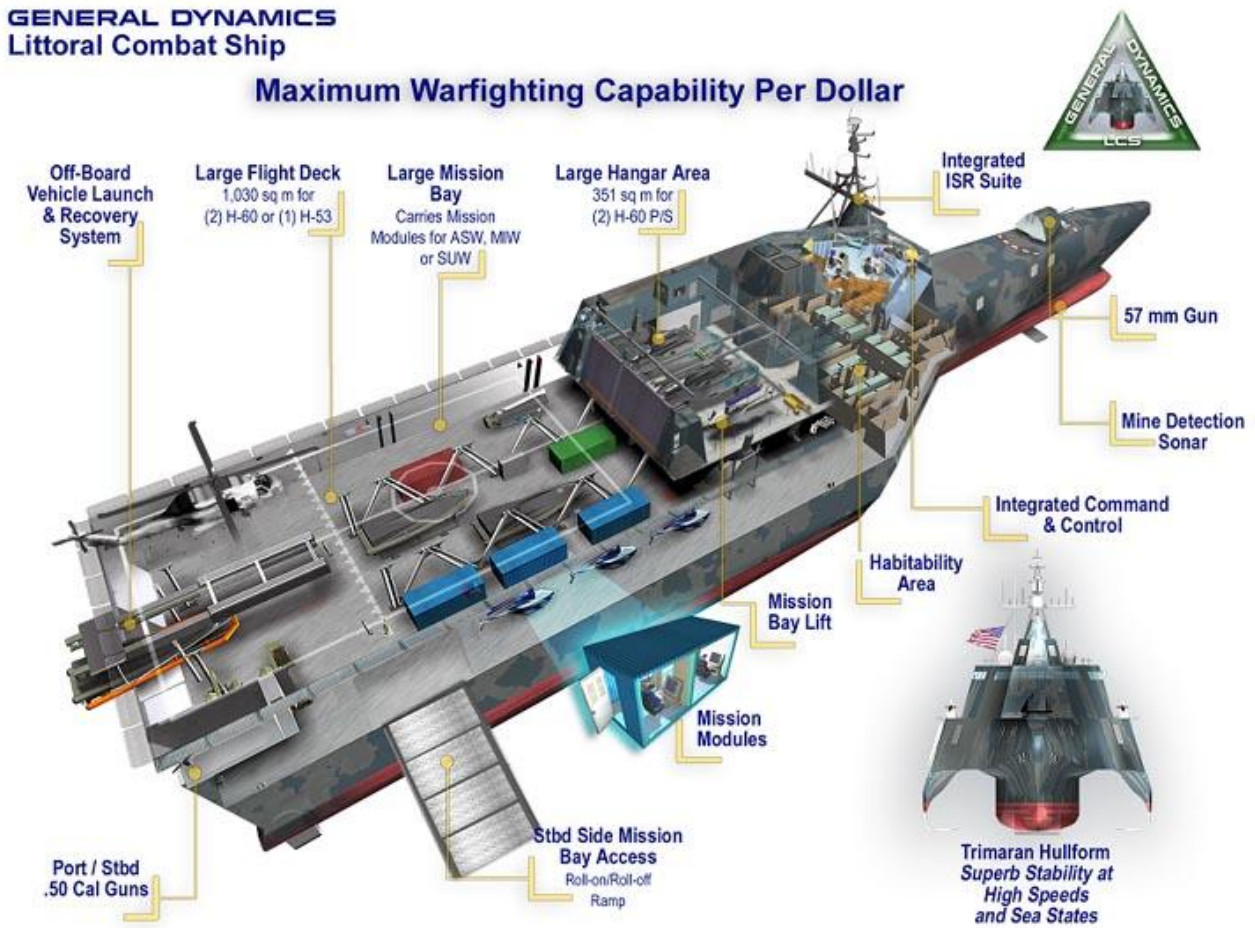
Maintainability

- **Analyzability**
  - *the ability to **identify the root cause** of a failure within the software.*
- **Changeability**
  - *Characterizes the **amount of effort** to change a system.*
- **Stability**
  - *Characterizes the **sensitivity to change** of a given system that is the negative impact that may be caused by system changes.*
- **Testability**
  - *Characterizes the effort needed to **verify** (test) a system change.*
- Maintainability Compliance

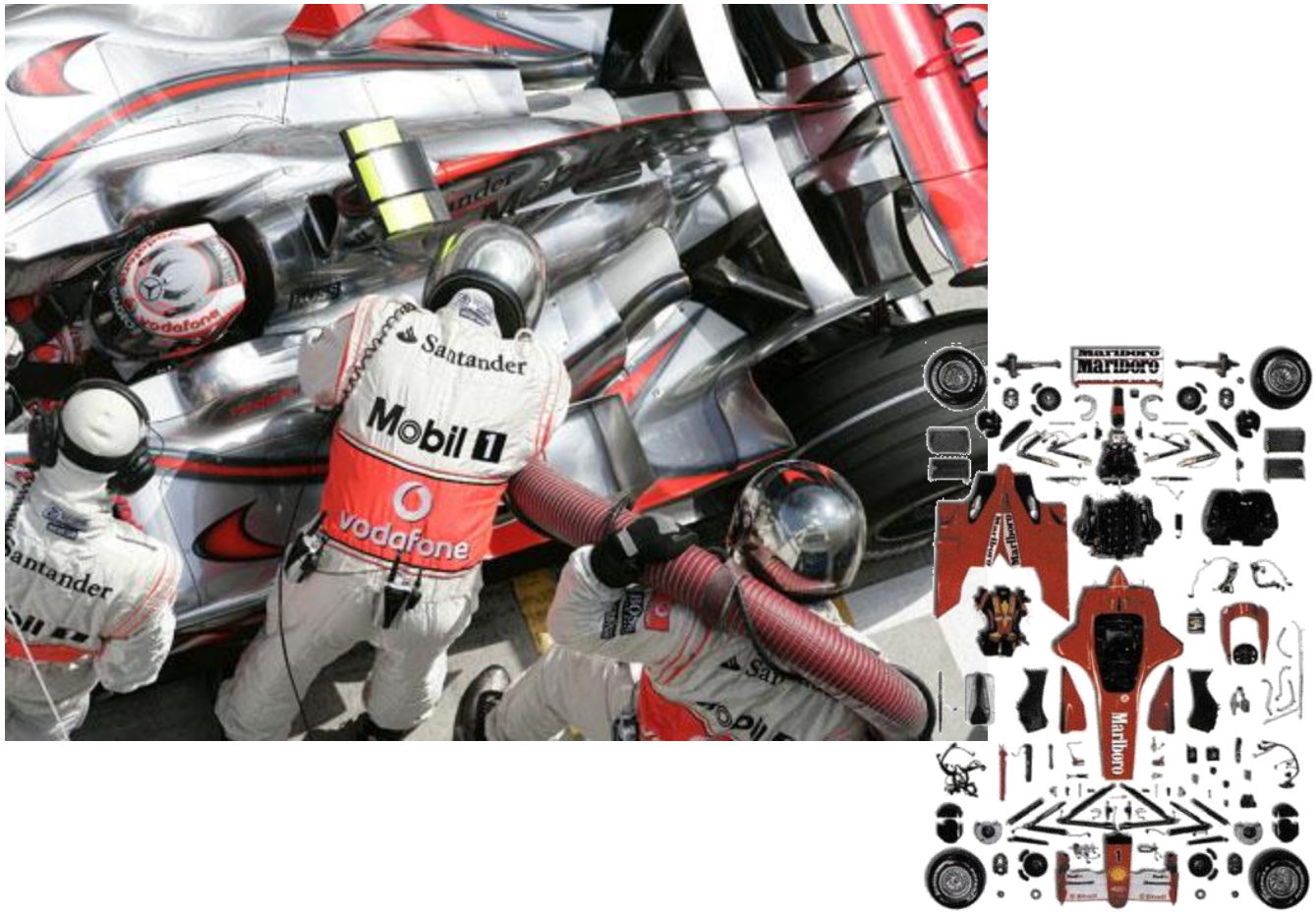
Portability

- **Adaptability**
  - *the ability of the system to **change to new specifications** or operating environments.*
- Installability
  - *the **effort** required to **install** the software*
- Co-Existence
  - *ability to **operate** several instances of the system with different software versions **in parallel***
- **Replacability**
  - the *plug and play* aspect of software components, that is how easy is it to **exchange** a given software **component** within a specified environment.
- Portability Compliance

Component designs

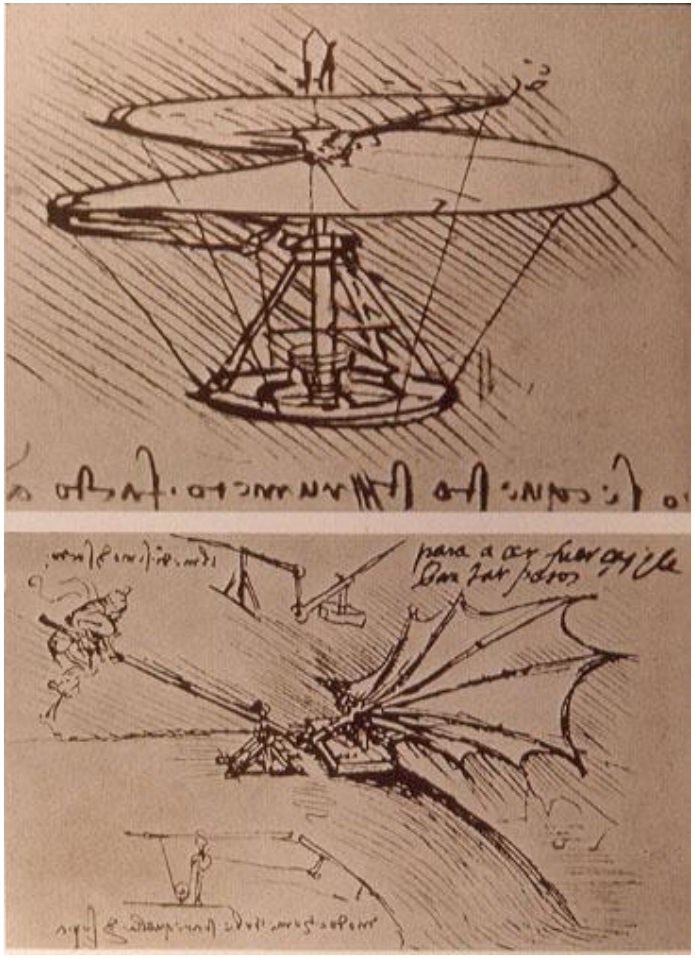


Performance & Optimisation

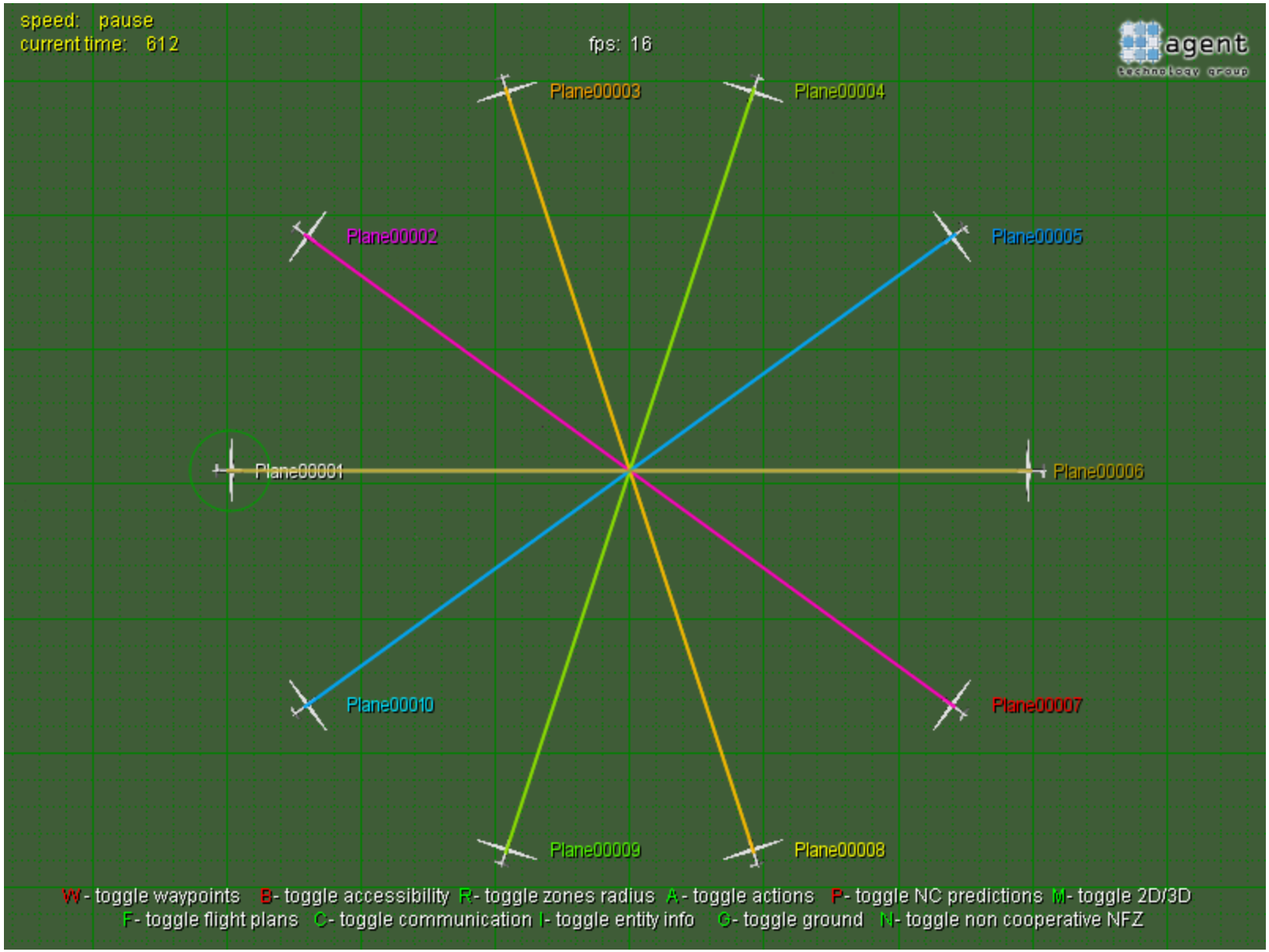


Performance & Optimisation

Design Patterns

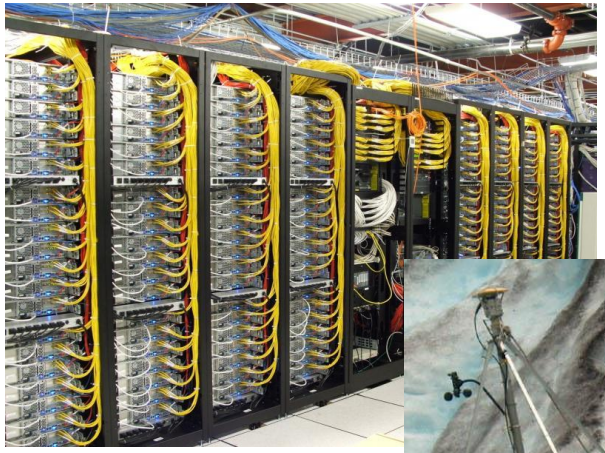


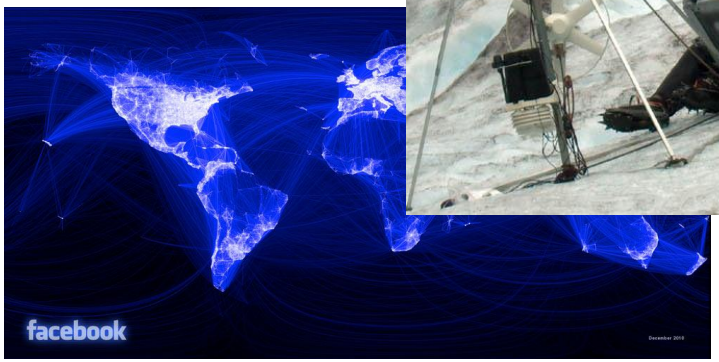
Concurrency



Distribution & Collaboration

Distribution & Collaboration





	Patterns & Components	Performance	Concurrency	Distribution& Collaboration
Maturity	●			
Fault Tolerance	●		●	
Recoverability	●			
Time Behavior		●	●	
Resource Utilization		●		
Scalability		●		●
Analyzability	●		●	
Changeability	●			●
Stability	●		●	●
Testability	●		●	●
Adaptability	●			●
Installability				●
Co-Existence	●			●
Replaceability	●			●