HMS Analytical Software

Coding for the enterprise server

What does it mean for you?

useR 2015 Conference Aalborg – Wednesday July 1, 2015

Friedrich Schuster

The „UniveRse“

Friedrich Schuster

Enterprise R

HMS Analytical Software

- Small, but growing part of the R universe
- R begins to replace first generation analytical platforms and solutions
- R has better price to performance ratio
- Data scientists want R
- Mission critical applications are ported to R

Questions:

- Is this possible and safe? Yes!
- Who can develop and support solutions?

Car insurance web portal

- To get an exact quote for your car you have to answer some questions about yourself, the car, and the requested insurance policy.
- The main challenges are: reliability (24/7), security and privacy, and of course you expect fast answers.

Quality control of consumables for diagnostic equipment

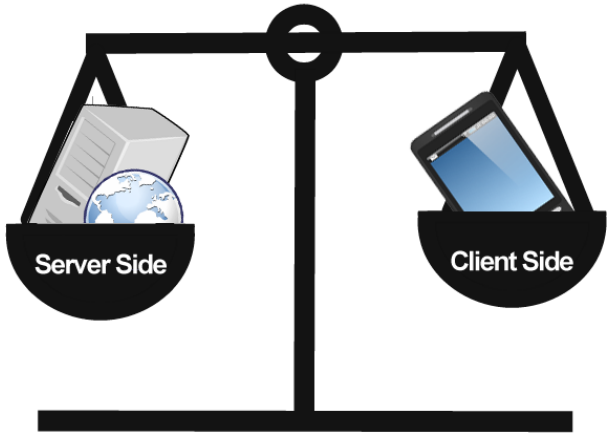
HMS Analytical Software

- Consumables for medical testing are a major source of income for a device manufacturer and have to be produced according to very high quality standards.
- Every batch is quality checked individually, and this process is supported by an server side application.
- Here the challenges are: stability (to avoid production downtime), and very strictly regulations set by the FDA.

Server side application characteristics

HMS Analytical Software

- Interaction via a separate **user interface**
- Multiple **sessions**
- Runs continuously: **stability**
- Does it work with any kind of **data**?
- **Results** are not guaranteed
- **Exception handling** and interactivity:
  What is an error? Who decides? How?



- **Messages**: with what content, in which language?
- **Division of labor**, roles, skills:
  Statisticians vs. developers vs. IT/operating
- **Software engineering, architecture**:
  High coupling? Bad! Separation of concerns? Good!

Corporate IT environments are special: The time factor

HMS Analytical Software

- Corporate IT environments are special: they have a long history ...
- ... and IT systems reflect this situation: even mainframes are still running.
- New applications have fit into this environment, and are expected to last 10 yers or more.
- Long term maintenance is expensive, so keeping the total cost (of ownership) for an application low has top priority within a company.

IT operates server applications

- Application owners vs. IT department & operating: **separate entities**

Requirements for maintaining an application:

- Stability, max. downtime, reaction time: **Service Level Agreements (SLA‘s)**
- Supported software versions: **stable, supported**
- **Emergency management**
- External **support, training** (e.g. Revo AdviseR, AcademyR)

Stable versions of R and packages / Reproducibility

HMS Analytical Software

- Required by regulations and/or IT operations
- In your own interest: stability, reliability of server application: use fixed / supported versions of **Open Source R**, **Revolution R Enterprise, etc.**
- Versioned Repository: **Revo Reproducible R Toolkit**
- Stable set of packages: **local repository (source, download)**, **Skript, installer or Linux package** with complete R and packages

Stakeholders and requirements

Most challenges are not directly related to the specific analytical task at hand.

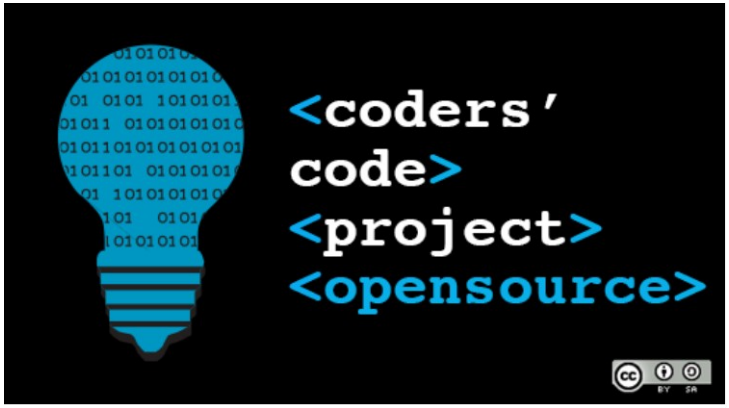
Many more challenges exist,

To make things more difficult: many stakeholders are involved in application related decisions, and all of them have their own priorities.

You may be the ones that have to make such decisions, or at least influence them as stakeholders

Well known tools and techniques

- Versioning: **Git**, **Subversion**; creating branches, merging branches, setting tags etc.
- Programming environments, IDEs: **Rstudio**, **Eclipse** etc.
- Unit-Testing: **Runit**, **Testthat**
- Documentation and reporting tools: **Roxygen, sweave, knitr, Markdown, pandoc**



Fault tolerant programming

- Handling exceptions: **try-catch**
- Raising exceptions, returning meaningful messages: **message(), warning(), stop()**
- Controlling **timeouts** and **loops**

Etc.

Internationalization

- Meaningful messages? In which language?
- Switching the locale of R **Sys.setLocale()** : not so good
- Throwing errors/exceptions with an **id number**
- Handling translation outside of R in the user interface or server application

Logging

HMS Analytical Software

- Logs: different reasons, types
  - Success of a specific sub-task (e.g. function call)
  - Complete user task or –session
  - Application log
  - Server log
- Laws and regulations: **traceability**
- Logging in a **single language**: English
- Different parallel user sessions: **separate logs** or **unique session identifier** in a common log
- **Logging configuration**: timestamps, source, messages, severity, target location
- Packages: **Logging**, **futile.logger**, **log4R**, **logR**, **loggr** etc.

Different (execution) environments

- Good practice: multiple, **separate development-, test- and production** environments
- Creating new instances: **installation packages, scripts, Linux packages**
- Installing / promoting application versions: **R packages**, configuration via **environment variables Sys.getenv()**, configuration files or DB
- Switching computing envionments: Revolution R Enterprise

Continuous integration (CI)

HMS Analytical Software

- Automates **repetitive tasks:** creation, installation and testing of software.
- Process is controlled by **build and test scripts.**
- Has it‘s own **web interface**
- **E.g. Jenkins**

Jenkins

Installation tests, functional tests

HMS Analytical Software

- Open Source R: **testInstalledBasic**(), **testInstalledPackages**()
- Revolution R Enterprise: **IQ/OP scripts**
- Functional qualification tests/scripts for custom packages
- Integration testing for the complete application?

Resource utilization / performance / scalability

- Save memory with **out-of-memory processing**: **Revo ScaleR**
- Gain performance with **in-memory** solutions, shared memory: **bigmemory**
- Flexibility in switching between in-memory, out-of-memory solutions: **Revo ScaleR**
- Switch between local and remote computing environments: **distributed computing**, **cloud computing**

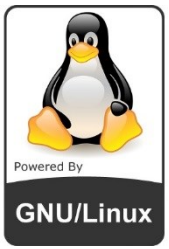
Isolation / Sandboxing / Containers

HMS Analytical Software

- Mandatory access control and sandboxing with **RAppArmor** from Jeroen Ooms (Linux only)
- Use **Docker** as application container (with sandboxing)?

Network / Server interfaces to R

- Language independent access/clients: **Rserve, DeployR, OpenCPU**
- Standards compliant interface: **Webservices**
- Webservices: **Revolution DeployR** / DeployR Open, **OpenCPU**, **R Service Bus**
- Flexibility, Scalability: **Revolution DeployR**
- For production better use: **Linux**
- Access LDAP, Active Directory from **Revo DeployR**, **Shiny Server Pro** etc



Please remember …

- server side applications are different for a reason.
- You should try to understand the specific situation and requirements of the (different) stakeholders and the IT department and the resulting requirements.

Only then should coding begin.

Thank you for your attention!
Any questions?

HMS Analytical Software

Friedrich Schuster
Senior Software Engineer

Contact:
HMS Analytical Software GmbH
Rohrbacher Str. 26 • 69115 Heidelberg
Telefon +49 6221 6051-142
Friedrich.Schuster@analytical-software.de

Web:
http://www.analytical-software.eu/