2nd CESSDA SaW Workshop
1-2 March 2017

CESSDA Technical Roadmap

John Shepherdson, UK
Data Service -
Head of CESSDA
Technical Working Group
Technical Working Group

Core group that aims to monitor and advise on needs regarding technical services.

Membership: CESSDA MO, DANS, DDA, FORS, FSD, GESIS, NSD, PROGEDO, SND, UKDS

See http://cessda.net/About-us/CESSDA-Working-Groups
Agenda

CESSDA’s Vision
- Overall
- Technical

Technical Roadmap
- 2015 - 2018

Technical Framework
CESSDA’s Vision

The vision of CESSDA, as stated in its statutes, is to provide a full scale sustainable research infrastructure that enables the research community to conduct high-quality research which in turn leads to effective solutions to the major challenges facing society today.
Technical Vision and Goals

To be recognized as a leading European Research Area (ERA) technical infrastructure

- Develop strong technology to support growth of and access to digital collections of SSH archives
  - metadata and datasets
- Develop, maintain and support software tools and technical infrastructure to support CESSDA’s overall goals
Research Infrastructure - 4 Pronged Approach

- People
- Technology
- Process
- Data/Information
Technical Roadmap 2015 - 2018

Past, present and future

Items in black are non-technical (included for completeness)
Aims to provide high-level view of CESSDA’s technical development work (past, present and future), so that readers can get a sense of what is in place, and what is to come on a year by year basis.
Technical Roadmap 2015

CESSDA Work Plan launched in 2015

- Open Source Metadata Harvester
- Metadata Management (phase 1)
- Technical Framework (phase 1)
- Expert Seminar on Trust (hosted by DANS)
- Data Access and Policy
Open Source Metadata Harvester

Reusable, extensible component
- Harvests metadata from Service Providers

Dependant on outputs from Metadata Management task

NSD + UKDS, FSD, SND, DDA
Completed June 2016
https://bitbucket.org/cessda/cessda.metadata.harvester
Metadata Management (phase 1)

Reusable Specifications

- standard metadata profile
- set of Controlled Vocabularies

FSD + ADP, CASD, DDA, GESIS, NSD, SND, UKDS
Ongoing to April 2017

CMM portfolio mandatory elements evaluation (by 10 March)

https://docs.google.com/spreadsheets/d/17jnXm9MCEjQcoboF-2VJ0kFPdL5Do2F0WTXup1k-LeFM/edit#gid=1896846293
Technical Framework (phase 1)

Building capability to maintain, enhance, QA, run technical Research Infrastructure

Specifications and infrastructure

- Technical Architecture document
- Shared development environment

UKDS + NSD, DANS, SND, GESIS
Completed April 2016
Technical Roadmap 2016

Work Plan tasks - new
- Technical Framework (phase 2)
- Products and Services Catalogue (phase 1)
- SP collaboration on Annex 6 GESIS tasks
  - Euro Question Bank
  - Training modules for ‘data discovery’
  - Persistent Identifier Services
- Expert Seminar re Technical Framework

Work Plan tasks - ongoing
- Metadata Management (phase 1)
Technical Framework (phase 2)

Specifications and infrastructure
- Technical Architecture document (revise)
- Shared development environment (extend)

CESSDA MO, DANS, FSD, FORS, GESIS, NSD, UKDS

Completed December 2016
Product and Services Catalogue (phase 1)

A one-stop-shop for search/discovery of datasets relevant to social science research

Pre-Development activities
- Requirements document
- User Interface wireframes

CESSDA MO, DANS, FSD, FORS, GESIS, NSD, UKDS

Completed December 2016
Euro Question Bank

Develop and implement a central ‘survey questions and variables’ search facility
  • Infrastructure plan
  • Requirements definition
  • User Interface design and testing

GESIS + FORS, FSD, NSD, SND, UKDS

Completed December 2016
Technical Roadmap 2017

Work Plan tasks - new
- Technical Framework (phase 3)
- Products and Services Catalogue (phase 2)
- Metadata Management (phase 2)
- Controlled Vocabulary Manager
Technical Roadmap 2017 (continued)

SP collaboration on Annex 6 tasks
- Euro Question Bank
- Training modules for Data Discovery Workshops and Collaborative data management
- Persistent Identifier Services

- CESSDA Expert Seminar 2017
- Trust Work Plan 2017 activities (SaW)
Technical Framework (phase 3)

Specifications and infrastructure
- Technical Architecture document (revise)
- Shared development environment (extend)

CESSDA MO, DANS, FSD, FORS, GESIS, NSD, UKDS, FSD, NA/DDA

Ends December 2017
Product and Services Catalogue (phase 2)

Development activities
- Make/Buy decision
- Implement ‘must have’ components

Ends December 2017

Virtual kick-off meeting March 2017
Metadata Management (phase 2)

Reusable Specifications
- extended metadata profile
- additional Controlled Vocabularies

Ongoing to December 2018
Euro Question Bank

- feedback survey & revised work plan
- data and metadata model
- second prototype and first productive application
- user workshops

Ends December 2017
Controlled Vocabulary Manager

Centralised management of Controlled Vocabularies that are used by many other systems

Specification (2017)

- Requirements gathering (inc. DDI Alliance)
- Survey existing tools
- Make/buy decision

Kick-off meeting at UKDA on 8/9 March 2017
Controlled Vocabulary Manager

Implementation (2018)
- Development
- Data population
- Testing
- Documentation

UKDS + FSD, GESIS, SND
Ends September 2018
Harmonisation

EQB and PaSC - compare UIs
EQB and CMM - check mappings from UI to model
PaSC and CMM - check mappings from UI to model

Need common approach for CESSDA UIs
Technical Roadmap 2018

Building on previous task outputs
- Technical Framework (phase 4)
- Product and Service Catalogue (phase 3)
- Controlled Vocabulary Manager (ongoing)
- Metadata Management (phase 2, ongoing)
- CESSDA Trust support activities
Technical Roadmap 2018

Additional proposals:
Authentication and Authorisation (inc SSO)
Data Archiving Tools
Secure Access Network
  • strategy, policy, tech requirements
Dataverse pilot(s) requirements
Vocabulary Service
Translation management platform
Exchange of services and support between SPs
  • including help with metadata harvesting
Impact analysis by MO
Vocabulary Service

Vocabulary Service (part of CESSDA Research Infrastructure)

- CVs
- Thesauri
- Ontologies
- ML Trans

CESSDA

Content Exporter

SP specific

Content Importer

ML Translation Platform

CESSDA systems

SP’s Systems

DDI Alliance Systems

SKOS RDF, JSON, XML, CSV

Ver 1.0 (2017-02-08)
Timetable for 2018 Workplan

- Spring 2017 - SPF reviews final package of proposals
- Spring 2017 - feasibility assessment by MO for financial and administrative issues
- Spring/Summer 2017 - package review by Board of Directors/MD
- Summer 2017 - present package to GA for approval
- January 2018 - start work
Other Technical Components

Number of projects produce components that COULD fit into CESSDA’s Research Infrastructure

- DASISH
- SaW
- SERISS
- Others

TWG will produce guidance for assessing and adopting/rejecting them
SERISS components

One example is SERISS* project

- Ends in 2019
- Produced number of relevant tools:
  - Questionnaire Design Documentation Tool (QDDT)
  - Question Variable Data Base (QVDB)
  - Survey Project Management Portal (SMAP)

*Synergies for Europe’s Infrastructures in the Social Sciences
Questions?
Technical Framework

Foundations for building and managing the technical infrastructure
Technical Framework at a glance

A guide for the development of the various (software) products and services that form part of the CESSDA Research Infrastructure

- Promote good practice for software development
- Meet common interoperability characteristics

Infrastructure for Development, Staging and Production

- Harmonise development tool chain for SPs
- Apply consistent set of tests
Technical Infrastructure for development

Cloud based development environment:
- Source code repositories, including
  - Issue Tracker
  - Wiki
- Continuous Integration and Testing

Staging and Production environments
- yet to be specified

Plus other one-off tools
Where to find it

Code repositories are in Bitbucket
Organised into projects
  • CRI (CESSDA Research Infrastructure)
  • CAG (CESSDA Architectural Guidelines)

Repository URLs
  https://bitbucket.org/cessda/<repo name>
E.g.
  https://bitbucket.org/cessda/cessda.guidelines.api
Research Infrastructure Repos

cessda.metadata.harvester
cessda.metadata.repository-handler.indexingplatform
cessda.metadata.repository-handler.nesstar
cessda.metadata.repository-handler.oai-pmh
cessda.metadata.repository-handler.omicrops
cessda.metadata.xslt-transformations.ddi32-flatten
cessda.repo.example
Architectural Guidelines Repos

Guidelines in CAG project

- API guidelines (cessda.guidelines.api)
- CIT guidelines (cessda.guidelines.cit)
  - For administrators
  - For Users
- Document templates (cessda.guidelines.doc-templates)
Protection of Software Assets

Ensure CESSDA has access to
• source code
• configuration files
• technical documentation

For Research Infrastructure components
Quality Control

Software Maturity Levels

• ensure quality of the research infrastructure is maintained
• guidance on minimum, expected and excellent standards
• originally based on NASA’s RRLs
• revised in light of ‘Capability Development Model’ from CESSDA SaW project
Maturity Modelling - More Info

See EDDI16 presentation:

A Capability Development Model for Assessing and Improving Distributed Infrastructures and their Services

Mike Priddy, Trond Kvakme, Marion Wittenberg
11 Product Acceptance Criteria

- Documentation
  - Development, Operational, End User
- Intellectual property issues
- Extensibility
- Modularity
- Packaging
11 Product Acceptance Criteria

- Portability
- Standards compliance
- Support
- Verification and testing
- Security
- Internationalisation and Localisation
Software Maturity Levels - SML

0 - Not applicable
1 - Initial usability; software use is not recommended
2 - Use is feasible; the software can be used by skilled personnel but with considerable effort, cost and risk
3 - Use is possible by most users; with some effort, cost, and risk. A risk assessment should be made before use
4 - Software is usable; with little effort, cost, and risk
5 - Demonstrable usability; there is clear evidence that the software is widely used by many users
1. Software developers have been identified and their responsibilities have been determined.

2. Developer organisation(s) (or developers) have an agreement with CESSDA that addresses any potential conflicts in the proposed intellectual property rights and responsibilities for development.

3. Agreements on development responsibilities, the list of developers, a recommended citation, and intellectual property rights statements, offering limited rights for use, are available, perhaps upon request, for review.
Intellectual Property

4. There is evidence that all developer organisation(s) (or developers) have confirmed that the list of developers, recommended citation, and intellectual property rights statements, including limited rights for use, in the software source code, documentation, and in the expression of the software upon execution, conform to CESSDA’s policies and agreements.

5. There is evidence that all developer organisation(s) (or developers) have confirmed that the list of developers, recommended citation, and intellectual property rights statements, including limited rights for use, in the software source code, documentation, and in the expression of the software upon execution, conform to CESSDA’s policies and agreements.
Intellectual Property

Developer sign up - get write access to CESSDA’s code repos

Complete online contributor’s agreement - re code ownership and IP

CESSDA Research Infrastructure Contributor License Agreement

CESSDA AS requires that You sign a Contributor License Agreement ("CLA") regarding any software code and/or documentation You wish to contribute to the CESSDA Research Infrastructure ("Contribution").

By submitting your Contribution to the CESSDA Research Infrastructure Project, You hereby agree to license your Contribution under the Apache License, Version 2.0 (https://www.apache.org/licenses/LICENSE-2.0), and to include the appropriate copyright notice required by the license.

*Required

- Given name(s) *
  e.g. Jane

- Family name *
  e.g. Smith

- Email address *

- Bitbucket account name *
# Software Maturity Levels Matrix

<table>
<thead>
<tr>
<th>Colour</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum standard</td>
<td></td>
</tr>
<tr>
<td>Expected standard</td>
<td></td>
</tr>
<tr>
<td>Excellent standard</td>
<td></td>
</tr>
</tbody>
</table>

The matrix details the progression of software development maturity levels, categorized under various aspects such as Development Environment, Testing, Review, and Security. Each level is color-coded to indicate its maturity level, with green representing excellent standards and red indicating minimum standards. The matrix provides a comprehensive overview of software development processes, highlighting critical areas for improvement at each level.
Online Form

https://goo.gl/forms/uwuye0nTUkti7AiH2

CESSDA Software Maturity Levels

Form v02.00, 23 September 2016

*Required

Using this form

You can use this form to assess a product (typically a 3rd party product you are thinking of adopting, or a component that is being considered for use as part of the CESSDA Research Infrastructure).

The information you enter will be stored in CESSDA’s Software Maturity Levels (SML) scorecard. When you reach the end of the form, you can choose to have the scores you entered emailed to you, along with the overall product assessment.

There are eleven criteria for you to score against. Each can be scored from Level 1 (low) to 5 (high), or 0 (not applicable).

Each of the criteria has a minimum standard associated with it, which should be met or exceeded by any component intended for use as part of the CESSDA Research Infrastructure.

Product name *

Your answer
Thanks for using the CESSDA Software Maturity Levels (SML) form.

The overall score for Open Source Metadata Harvester (OSMH) is Level 3

Which means "Software is usable; the software can be used by most users although there may be some cost and risk" (i.e. expected standard for CESSDA RI use).

You entered the following values:

- CA1.1: End user Documentation 0
- CA1.2: Operational documentation 2
- CA1.3: Development Documentation 2
- CA2: Intellectual Property 3
- CA3: Extensibility 4
- CA4: Modularity 3
- CA5: Packaging 2
- CA6: Portability 4
- CA7: Standards Compliance 2
- CA8: Support 2
- CA9: Verification and Testing 3
- CA10: Security 3
- CA11: Internationalisation and Localisation 3

and provided the following feedback:

"It would sometimes be good to be able to comment on a score, because there are nuances and trade-offs here. Also, an N/A-category could be useful. Used 0 for the CA1.1 since the OSMH don't have "end users" as such."

Regards,
The CESSDA Technical Work Group
Summary/Review
Deep dive: CIT

CIT (Continuous Integration and Testing) is an approach that builds and tests a software product whenever changed source code is committed to the Version Control System.

Use ‘web hook’ to connect CI server to VCS

=> Product may be built and tested many times per day.
CIT basics

- Instrumentation
- Unit testing
- Style checking
- Documentation
CIT instrumentation

<table>
<thead>
<tr>
<th>Job Status</th>
<th>Build History</th>
</tr>
</thead>
</table>

![Jenkins Build History](image)

<table>
<thead>
<tr>
<th>Jenkins Job Status</th>
<th>Jenkins Build History</th>
</tr>
</thead>
</table>

**Build Queue**
No builds in the queue.

**Build Executor Status**

<table>
<thead>
<tr>
<th>#</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Idle</td>
</tr>
<tr>
<td>2</td>
<td>Idle</td>
</tr>
<tr>
<td>3</td>
<td>Idle</td>
</tr>
<tr>
<td>4</td>
<td>Idle</td>
</tr>
</tbody>
</table>

**Jenkins Build History**

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>Name</th>
<th>Last Success</th>
<th>Last Failure</th>
<th>Last Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Citizen portal check</td>
<td>0 min 56 sec (1511)</td>
<td>N/A</td>
<td>1 min 6 sec</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>UKDA Status Service API</td>
<td>24 min (1282)</td>
<td>N/A</td>
<td>16 sec</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>sched stormtraineventupdate</td>
<td>53 min (1511)</td>
<td>N/A</td>
<td>2.4 sec</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>sched stormtraineventupdate</td>
<td>53 min (1511)</td>
<td>N/A</td>
<td>2 sec</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Nextor portal check</td>
<td>1 hr 29 min (1429)</td>
<td>N/A</td>
<td>31 sec</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>UKDA Utility Library</td>
<td>2 hr 11 min (1429)</td>
<td>N/A</td>
<td>45 sec</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>UKDS Unblock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>sched admincenter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>store portal check</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>UKDA NewestEvents</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Project UKDA.API.Service**

![Project UKDA.API.Service](image)

**Upstream Projects**
UKDA.API.Data

**Permalinks**
- Last build (#162), 7 days 2 hr ago
- Last successful build (#159), 7 days 9 hr ago
- Last failed build (#159), 7 days 22 hr ago
- Last upstream build (#159), 7 days 22 hr ago
CIT instrumentation

Build artefacts

Revision history

Audit trail

Build #20 (Feb 23, 2010 11:38:29 AM)

Build Artifacts
- CommonUtilities_1_1.jar
- Configuration_en_GB.properties
- log4j-1.2.15.jar
- log4j.properties
- Message_en_GB.properties
- ReadMe.txt

Revisions
- http://dadevapp/svn/JavaBuildAndTest/trunk: 10

No changes.

Started by user Dovao, Oscar
CIT unit testing

Test Result Trend

Code Coverage Trend

Test Result: UnitTests

0 failures (40)

All Tests

<table>
<thead>
<tr>
<th>Test name</th>
<th>Duration</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cookies_GetScriptAndStyleTagOutput_ReturnsEmptyString_WhenCookieAccepted</td>
<td>0.16 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>Cookies_GetScriptAndStyleTagOutput_ReturnsString_WhenCookieNotAccepted</td>
<td>0.25 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>Cryptography_EncryptSetValue_OutOfRangeThrowsException</td>
<td>11 ms</td>
<td>Passed</td>
</tr>
<tr>
<td>Cryptography_Encrypt_Decrypt_WithDefaults_RoundTrips</td>
<td>0.19 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>Cryptography_Encrypt_Decrypt_WithHashAlgorithmMds_RoundTrips</td>
<td>14 ms</td>
<td>Passed</td>
</tr>
<tr>
<td>Cryptography_Encrypt_Decrypt_WithHashAlgorithmRsa1_RoundTrips</td>
<td>24 ms</td>
<td>Passed</td>
</tr>
<tr>
<td>Cryptography_Encrypt_Decrypt_WithHashAlgorithmRsa2_RoundTrips</td>
<td>12 ms</td>
<td>Passed</td>
</tr>
<tr>
<td>Cryptography_Encrypt_Decrypt_WithHashAlgorithmSha256_RoundTrips</td>
<td>5 ms</td>
<td>Passed</td>
</tr>
</tbody>
</table>
CIT source code style checking

Violations Report for build 275

<table>
<thead>
<tr>
<th>Type</th>
<th>Violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>stylecop</td>
<td>72</td>
</tr>
</tbody>
</table>

Stylecop CheckStyle

```
> using System;
> using System.Collections.Generic;
> using System.Linq;
> using System.Text;
> using HtmlAttributes;
> namespace DataObjects.Revu.Entities
{
    public class RevuRecord
    {
        Type   | Class               | Description                                      |
        stylecop | DocumentationRules  | The class must have a documentation header. (SA1600) |
    }
```

<table>
<thead>
<tr>
<th>filename</th>
<th>l</th>
<th>m</th>
<th>h</th>
<th>number</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommonUtilities/src/main/com/ukda/common/ConfigurationHandler.java</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1 (+1)</td>
</tr>
<tr>
<td>CommonUtilities/src/main/com/ukda/common/DateUtil.java</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1 (+1)</td>
</tr>
<tr>
<td>CommonUtilities/src/main/com/ukda/common/DebugHandler.java</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1 (+1)</td>
</tr>
</tbody>
</table>
CIT source code QA

Report from Sonar plugin

<table>
<thead>
<tr>
<th>Alert</th>
<th>Name</th>
<th>Version</th>
<th>Lines of code</th>
<th>Rules compliance</th>
<th>Build date</th>
</tr>
</thead>
<tbody>
<tr>
<td>SkosHassetGenerator</td>
<td>trunk</td>
<td>107</td>
<td>90.7%</td>
<td></td>
<td>23 Jan 2013</td>
</tr>
<tr>
<td>UKDA API Data</td>
<td>trunk</td>
<td>14,326</td>
<td>90.4%</td>
<td></td>
<td>10 Apr 2014</td>
</tr>
<tr>
<td>UKDA API Service</td>
<td>trunk</td>
<td>15,494</td>
<td>92.7%</td>
<td></td>
<td>04 Sep 2013</td>
</tr>
<tr>
<td>UKDA Search Library</td>
<td>trunk</td>
<td>10,115</td>
<td>96.3%</td>
<td></td>
<td>02 Apr 2014</td>
</tr>
<tr>
<td>UKDA Search MVC</td>
<td>trunk</td>
<td>8,536</td>
<td>100.0%</td>
<td></td>
<td>05 Feb 2013</td>
</tr>
<tr>
<td>UKDA Utility Library</td>
<td>trunk</td>
<td>3,056</td>
<td>99.9%</td>
<td></td>
<td>11 Feb 2013</td>
</tr>
</tbody>
</table>
CIT generating database documentation

Report from Red Gate SQL Doc tool
CIT generating database documentation

Report from Red Gate SQL Doc tool

<table>
<thead>
<tr>
<th>Columns</th>
<th>Data Type</th>
<th>Max Length (Bytes)</th>
<th>Allow Nulls</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>uniqueidentifier</td>
<td>16</td>
<td>✗</td>
<td>(newsequentialid())</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Indexes</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Columns</td>
<td>Unique</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PK__LANGUAGE__3214EC2700551192</td>
<td>ID</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNQ__Language__TwoCode</td>
<td>TWOCODE</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNQ__Language__ThreeCode</td>
<td>THREECODE</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Here are the classes, structs, unions and interfaces with brief descriptions:

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileProcessors.DirectoryCopier</td>
<td>Copies the contents of a source directory into a target directory, including directories and files.</td>
</tr>
<tr>
<td>Extensions.DirectoryInfoExtensions</td>
<td>Reads a PDF update profile from configuration files.</td>
</tr>
<tr>
<td>FileProcessors.PdfUpdaters.TabulaDX.SRN.PdfProfiles</td>
<td>Copies the contents of a source directory into a target directory, including directories and files.</td>
</tr>
<tr>
<td>FileProcessors.PdfUpdaters.PdfUpdater</td>
<td>Copies the contents of a source directory into a target directory, including directories and files.</td>
</tr>
<tr>
<td>FileProcessors.PdfUpdaters.PdfUpdater</td>
<td>Copies the contents of a source directory into a target directory, including directories and files.</td>
</tr>
<tr>
<td>SRNTools.Program</td>
<td>A strongly-typed resource class, for looking up localized strings, etc.</td>
</tr>
<tr>
<td>SRNTools.Properties.Resources</td>
<td>Modifies the names of subdirectories and metadata field names into for PDF update profiles.</td>
</tr>
<tr>
<td>SRNTools.SRNToolsForm</td>
<td>Extensions to the System.IO.DirectoryInfo object.</td>
</tr>
</tbody>
</table>

Report from dOxygen tool
CIT User journey QA

Selenium - plugin for Firefox
- Allows User clicks to be captured
  - convert to test scripts
  - run on demand

- Selenium Grid
  - can test multiple browser types and variants
CIT User journey QA

Using Selenium plugin

<table>
<thead>
<tr>
<th>Test name</th>
<th>Duration</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>QualiBankAgeFilter</td>
<td>33 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankAutoComplete</td>
<td>13 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankClearSearch</td>
<td>13 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankCollectionFilter</td>
<td>25 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankDateFilter</td>
<td>26 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankFilterCoreFacet</td>
<td>24 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankFilterPopup</td>
<td>22 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankGenderFilter</td>
<td>24 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankHighlight</td>
<td>11 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankPagination</td>
<td>11 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankPopup</td>
<td>22 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankQuickSearch</td>
<td>11 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankRegionFilter</td>
<td>25 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankResetFilter</td>
<td>36 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankResourceFilter</td>
<td>24 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankSearchResult</td>
<td>11 sec</td>
<td>Passed</td>
</tr>
<tr>
<td>QualiBankSocioeconomicFilter</td>
<td>25 sec</td>
<td>Passed</td>
</tr>
</tbody>
</table>
CIT User journey QA

Using Selenium plugin
Designing and Building for the Cloud
Heroku’s 12 Factors (see 12factor.net)
Guidelines for “cloud native” development

Why “cloud native”??
Factor 1 - Codebase under version control

One Codebase - One Application

(Content courtesy of Ørnulf Risnes, NSD)
Factor 2 - Explicitly declare and isolate dependencies

Declare everything

Trust no one

(Content courtesy of Ørnulf Risnes, NSD)
Factor 3 - Configuration (and credentials)

Configuration is the only difference between the deploys below

(Content courtesy of Ørnulf Risnes, NSD)
Factor 4 - Backing Services

Treat everything (including File System) as a service

(Content courtesy of Ørnulf Risnes, NSD)
Factor 5 - Design, Build, Release, Run

Fully automated and iterative pipeline

(Content courtesy of Ørnulf Risnes, NSD)
Factor 6 - Applications are stateless

No state?

State is delegated to backing services (dbs, file storage, caches).

(Content courtesy of Ørnulf Risnes, NSD)
Factor 7 - Apps are self-contained servers with *externally* configurable ports

Server is a part of the app

Ports can be managed from outside

(Content courtesy of Ørnulf Risnes, NSD)
Factor 8 - Concurrency

Stateless apps enable horizontal scaling

Re: The Scale Cube X-axis

(Content courtesy of Ørnulf Risnes, NSD)
The Scale Cube - Scalable architecture

Y axis - functional decomposition
Scale by splitting different things

X axis - horizontal duplication
Scale by cloning

Z axis - data partitioning
Scale by splitting similar things

(Content courtesy of Ørnulf Risnes, NSD)
Factor 9 - Disposability

Apps are ephemeral
Graceful shutdown
Fast startup
  Limit warm-up, delegate to backing cache

(Content courtesy of Ørnulf Risnes, NSD)
Factor 10 - Dev/prod parity

“But it works on my machine!”

Reduce gap
Time (deploy frequently)
People (deploy automatically)
Resources (use prod-like tools also for devs)

(Content courtesy of Ørnulf Risnes, NSD)
Factor 11 - Logs

Log to stdout and stderr

Log aggregation, storage, processing part of the runtime platform

(Content courtesy of Ørnulf Risnes, NSD)
Factor 12 - Admin processes for apps

N/A if your app is stateless and configuration is immutable

(Content courtesy of Ørnulf Risnes, NSD)
Thank you for listening

Any questions?
cessda
—in your circle of trust.

website: www.cessda.net / twitter: @CESSDA_Data