



Continuous Delivery in the Real World

Or how we moved away from forming our bricks manually

Continuous Delivery in the Real World

Or how we moved away from forming our bricks manually

Harald Göttlicher, Software Architect
Robert BOSCH GmbH
Automotive Aftermarket

- What do we do?
- Where do we come from?
- What did we change?
- How do we use Jenkins?

What do we do?

ESI[tronic] Automotive Diagnostics solutions for the workshop

- ➔ Diagnostic Software Applications
- ➔ Troubleshooting instructions, Circuit Diagrams, Spare Parts Lists, ...
- ➔ Soft- and Hardware
- ➔ Support and Services

ESI[tronic] Software

- ➔ ~65.000 customers
- ➔ PC and embedded Linux
- ➔ 3-DVD delivery 3x/year
- ➔ online updates ~weekly

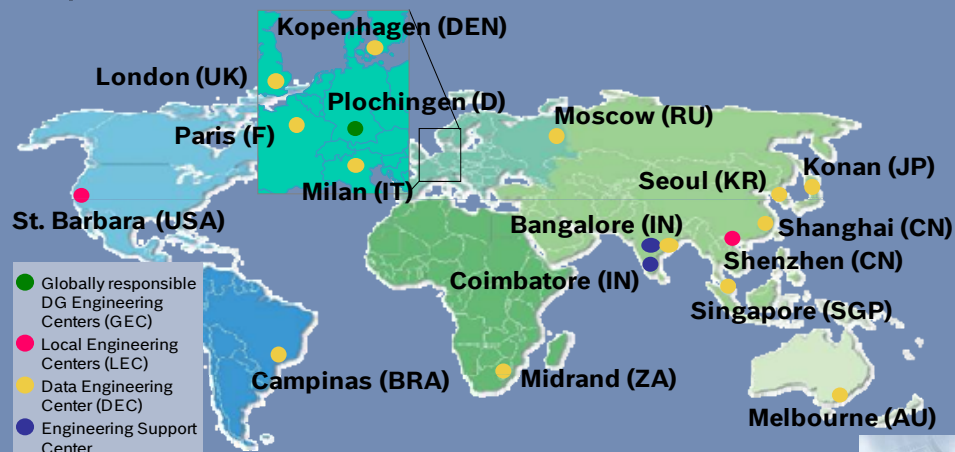
www.esitronic.com



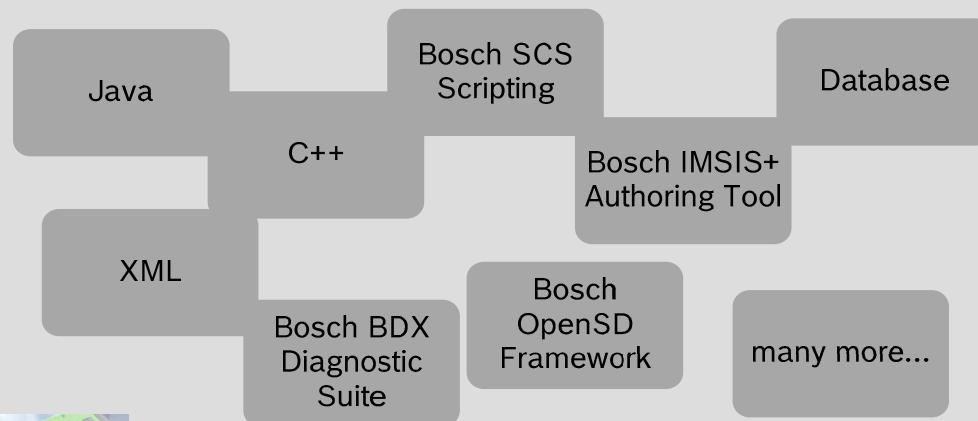
Continuous Delivery in the Real World

Why is it complex to build it?

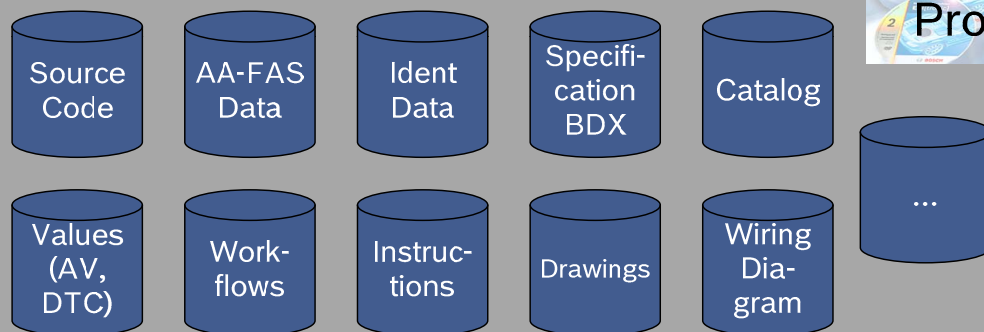
big worldwide team: 18 sites, >500 developers



For runtime: >20 technologies because of 15 years of legacy

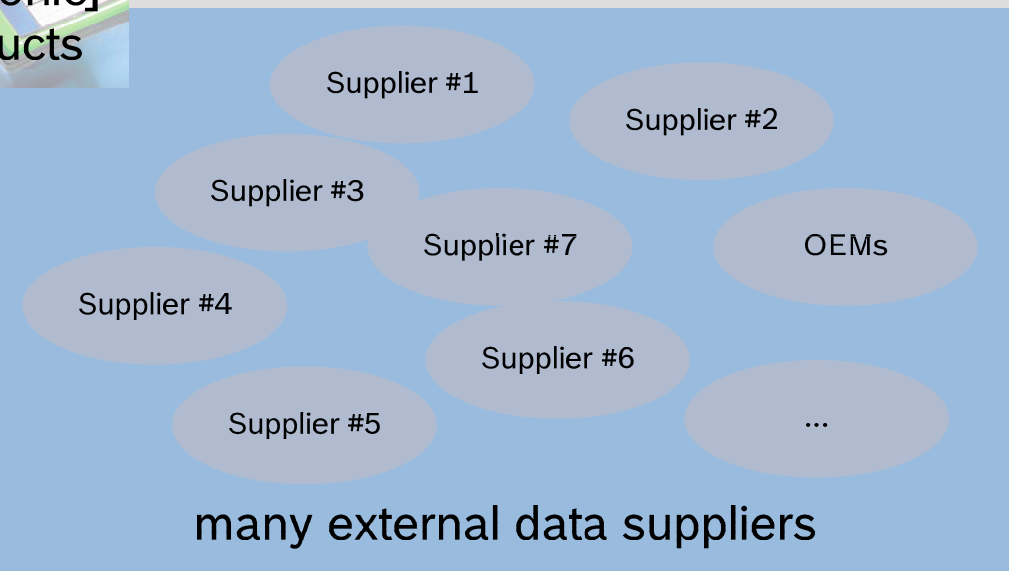


ESI[tronic]
Products



>10 data types, big data, many dependencies

For runtime: 600.000 files - 15 GB (3 dual layer DVDs)



Where do we come from?

15-year history of a Windows 3.11 plus port to Linux-embedded leads to

- grown heterogeneous build landscape
- growing needs created many “private” building places
- only manually interacting, many manual steps

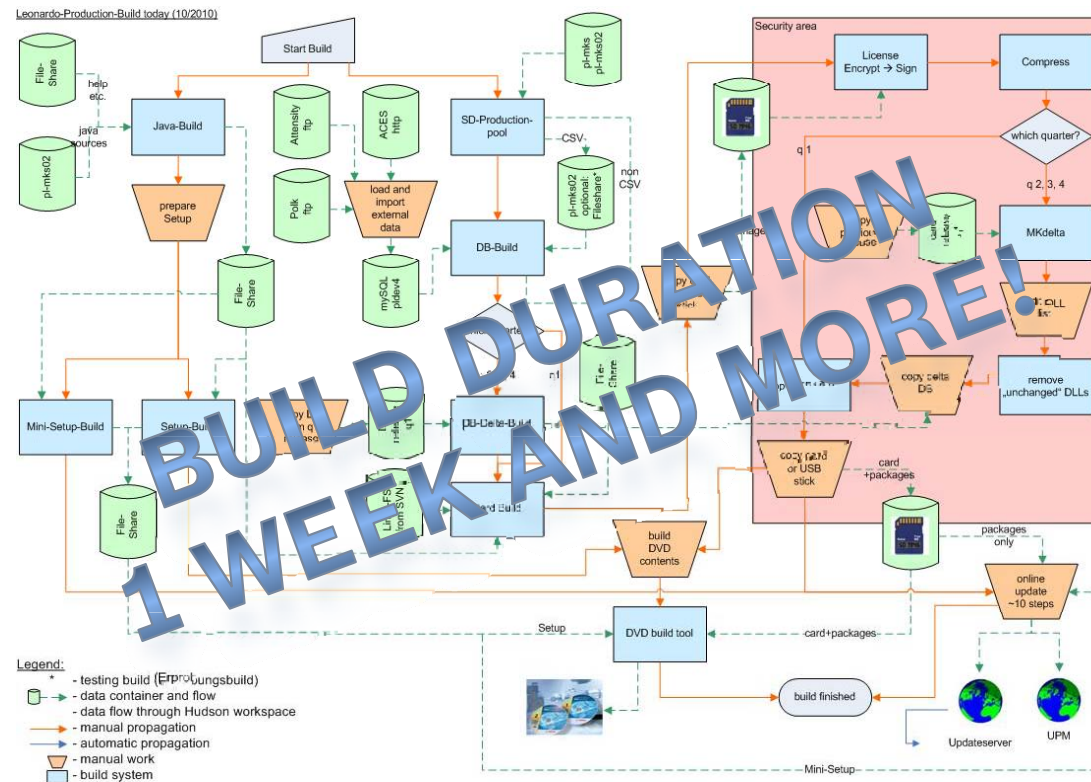
Overview was decreasing.



Where do we come from?

Grown fragmented build landscape leads to

- complex, error-prone manual workflow (orange arrows)
- lots of manual work (orange blocks)
- risk of errors, delays and cycles



Where do we come from?

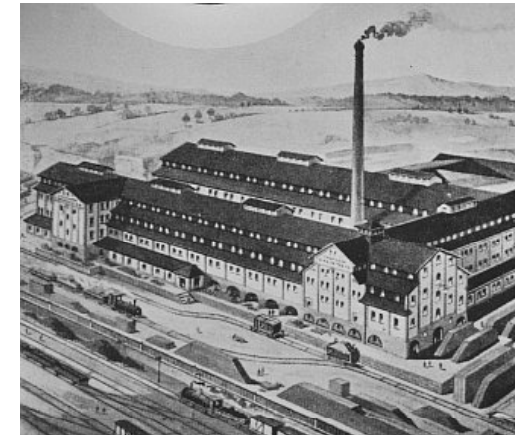
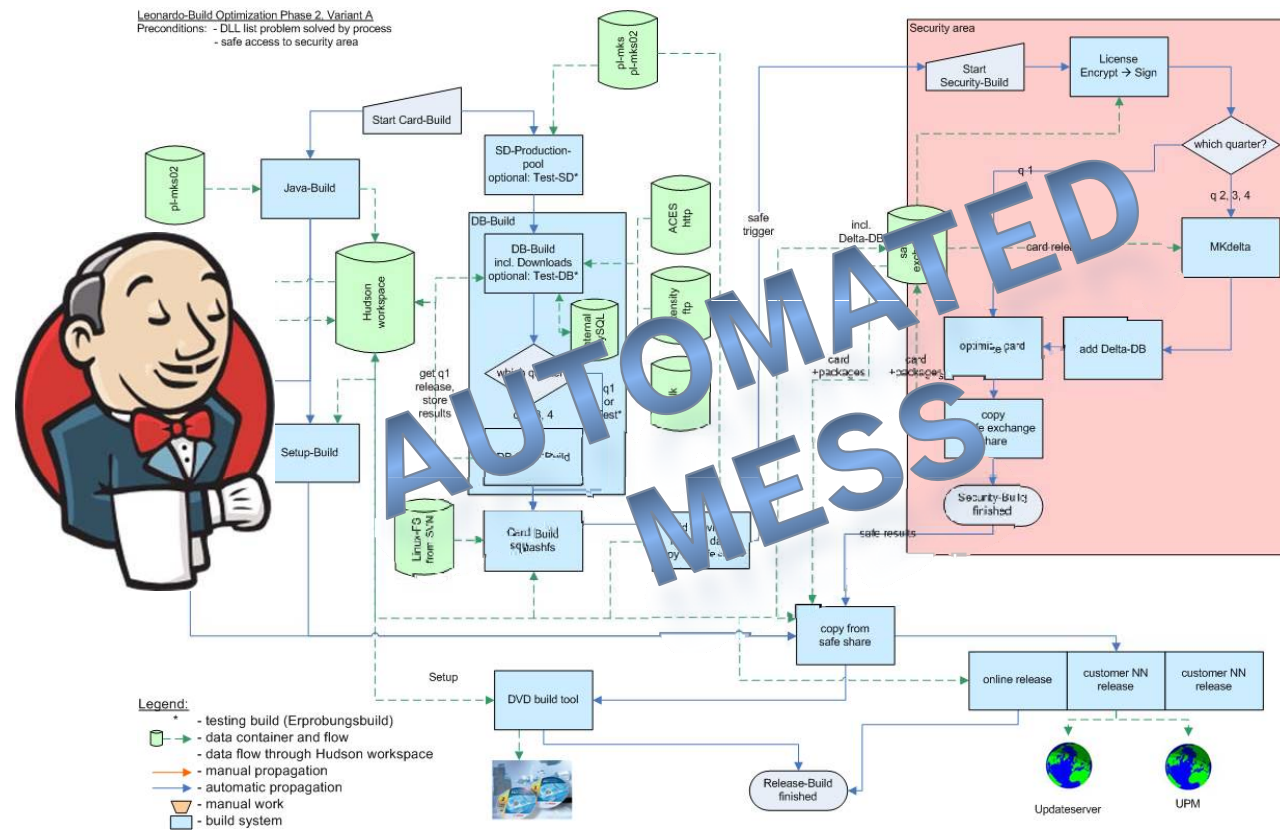
- legacy mix of outdated technologies – partially even instable
- some developed and operated by single persons
- know-how was spread



What did we change?

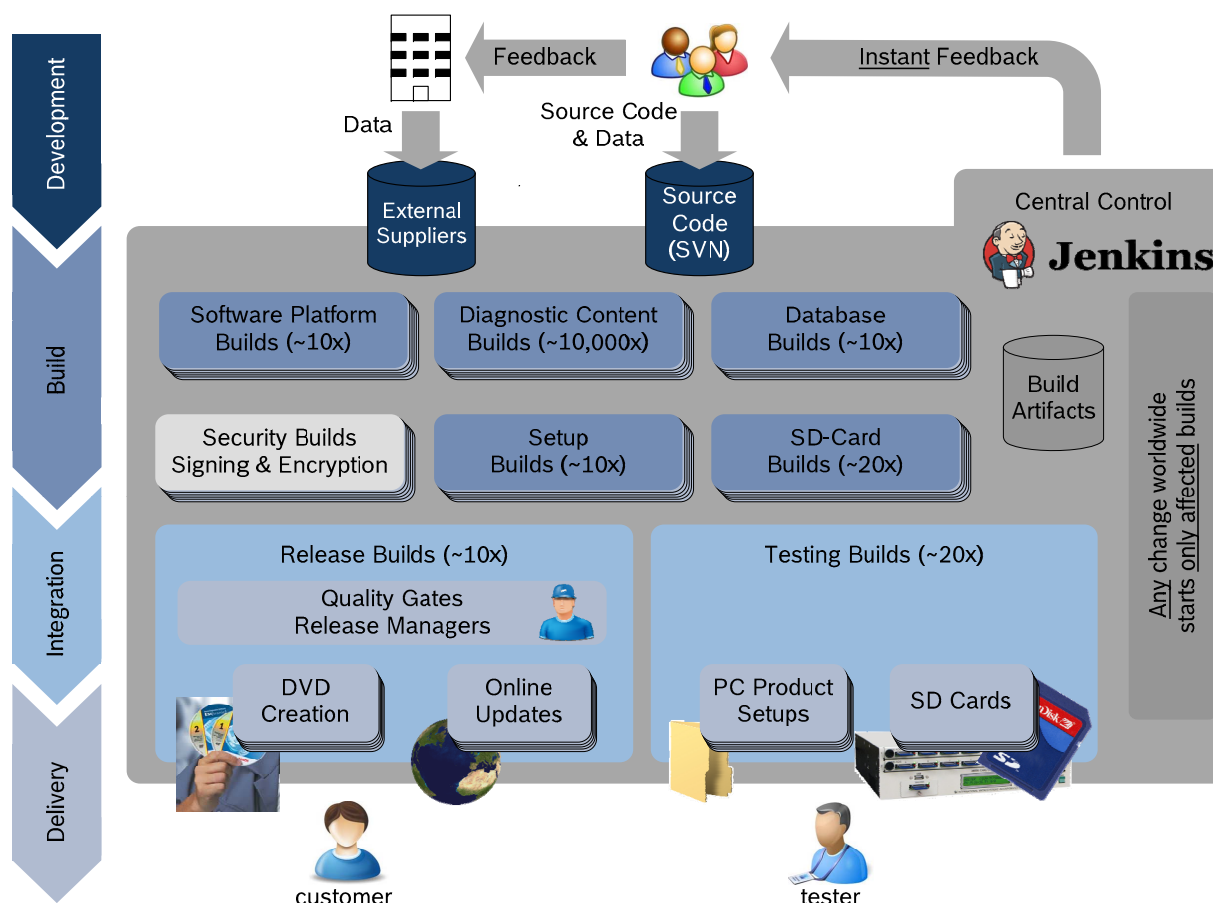
- ➔ First approach: connect and automate the existing builds with Jenkins
- ➔ plus continuous integration of source code

If you have a mess and you automate it, you have an automated mess!



What did we change?

- Split up Software in up to 10,000s of small artefacts!



- What are the artefacts **you** are producing?



What did we change?

Align build tools step by step to

- Jenkins
- Gradle
- Artifactory
- Linux-Bash



What did we change?

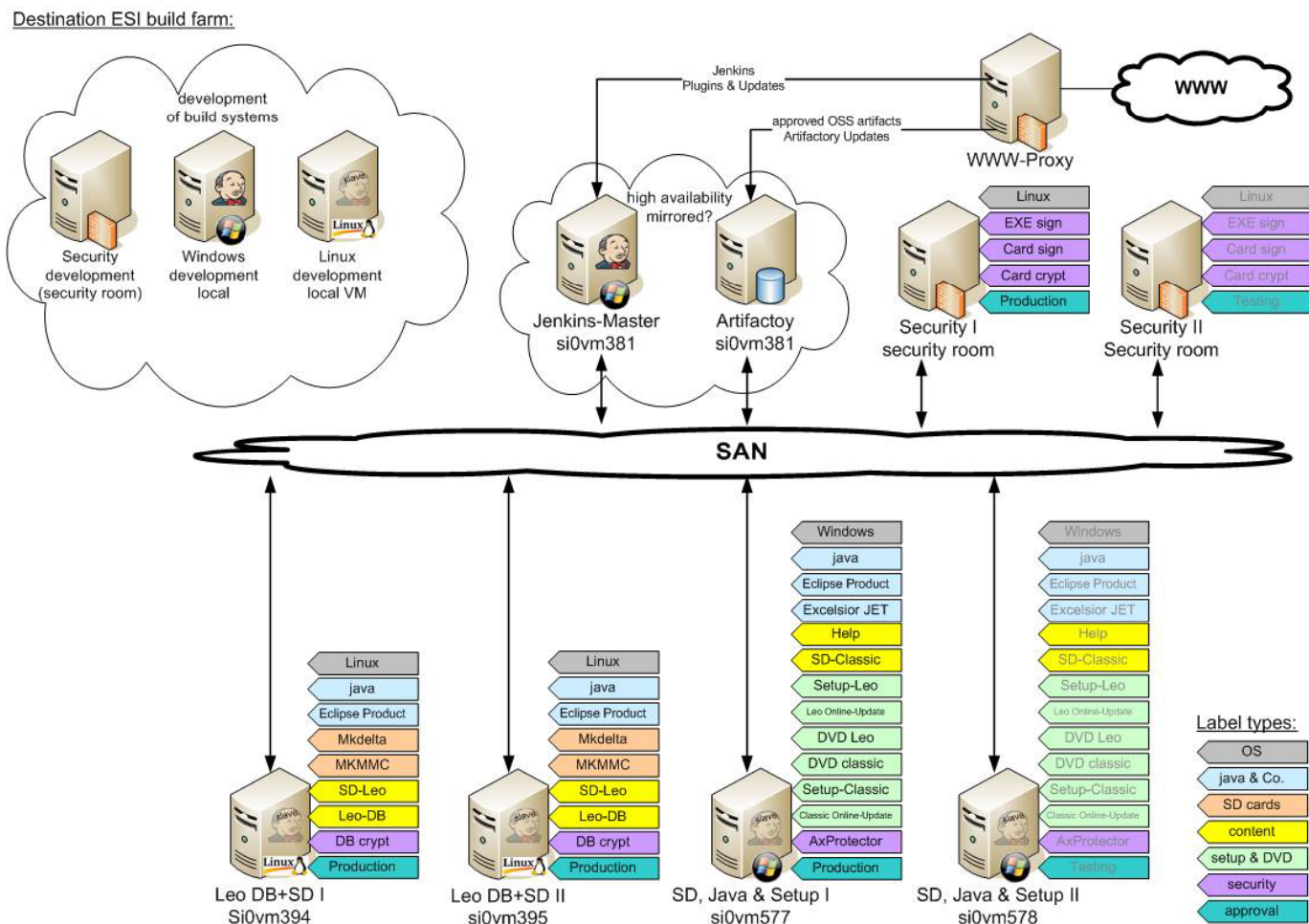
Custom tools for developers

- Automated runtime update from Artifactory
- Ease of source commit from runtime
- One-click creation of “checkpoints” on Artifactory
- One-Click creation of new projects / branches



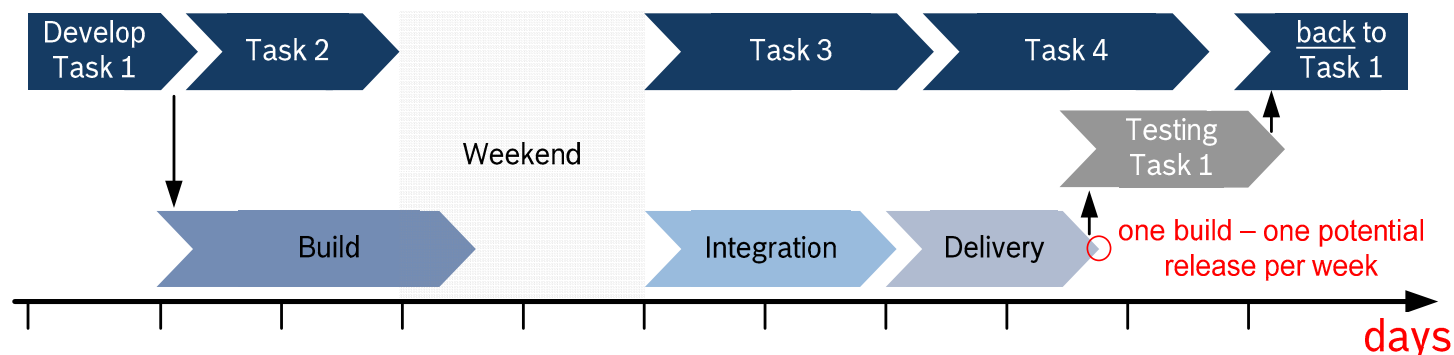
What did we change?

- Unify builds and Build Servers for Load Distribution / Redundancy

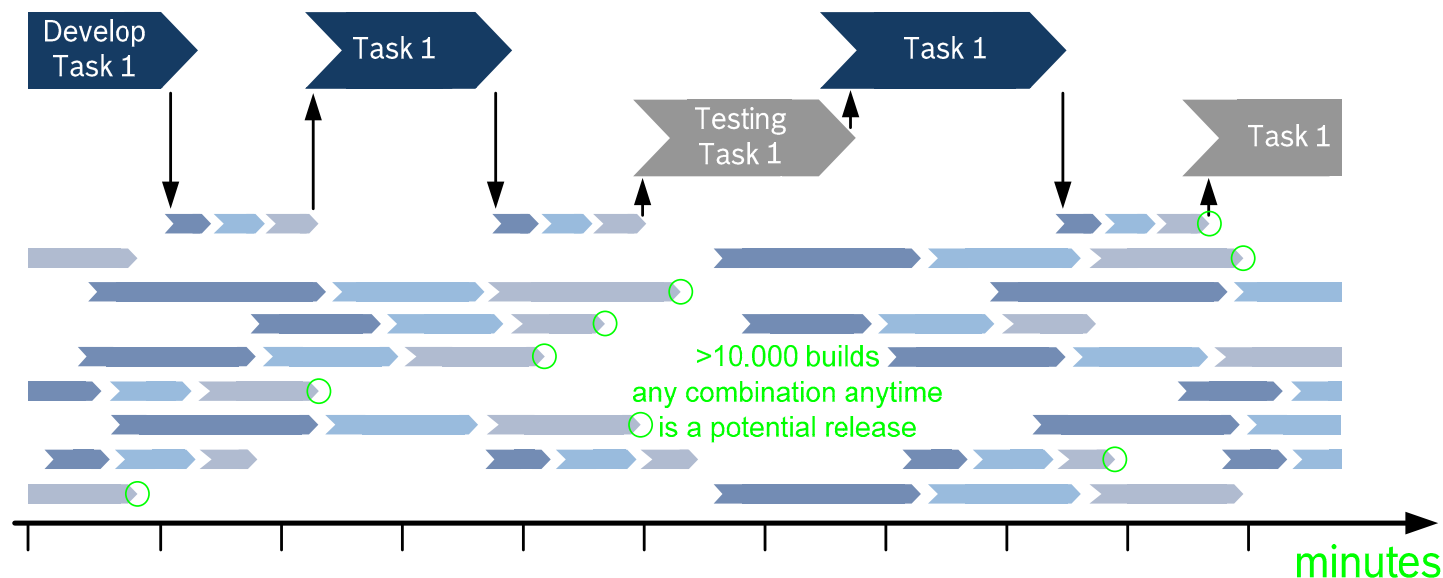


What did we change?

2011: Full builds ~ 1 week – Developer and Tester not focused!



2013: Feedback within Minutes – Developer and Tester focused on one task



Highlights

- Higher quality by instant testing of small packages
- Enables agile development and high development speed
- Faster feedback cycle to customer
- Prerequisite for daily updates to customer

What did we change?



What's next?

- Go live of remaining parts
- Unify some more old systems
- Extend Continuous inspection with SonarQube
- Connect and extend test automation



How do we use Jenkins?

How to set up Jenkins for such a challenging environment?

- How to deal with 10,000s of builds?
- How to deal with many projects and users?
- How to deal with special requirements?
- Is it scalable / stable enough?

→ Let's get technical now!



How do we use Jenkins?

Challenge #1: Creating and maintaining 10,000s of Build Projects

- Use templates plug-in to
 - Create build projects programmatically
 - Restrict editable parameters
 - Maintain / change all build projects at once
- Live Demo...



Continuous Delivery in the Real World

How do we use Jenkins?

Templates plug-in: Screenshots from Live Demo...

The screenshot shows the Jenkins job configuration page for 'SD Main Branch trunk'. The left sidebar contains links for 'Configure', 'Move', 'Groups', 'Roles', 'Email Template Testing', 'Subversion Polling Log', 'Build History', 'Workspace', and 'Recent Changes'. The main content area shows the job configuration, including a 'Transformer' section with a Groovy script for template transformation. The script defines variables for repository URL, source base path, component type name, and job type arguments. It also includes logic for handling different branches and a boilerplate for the job configuration.

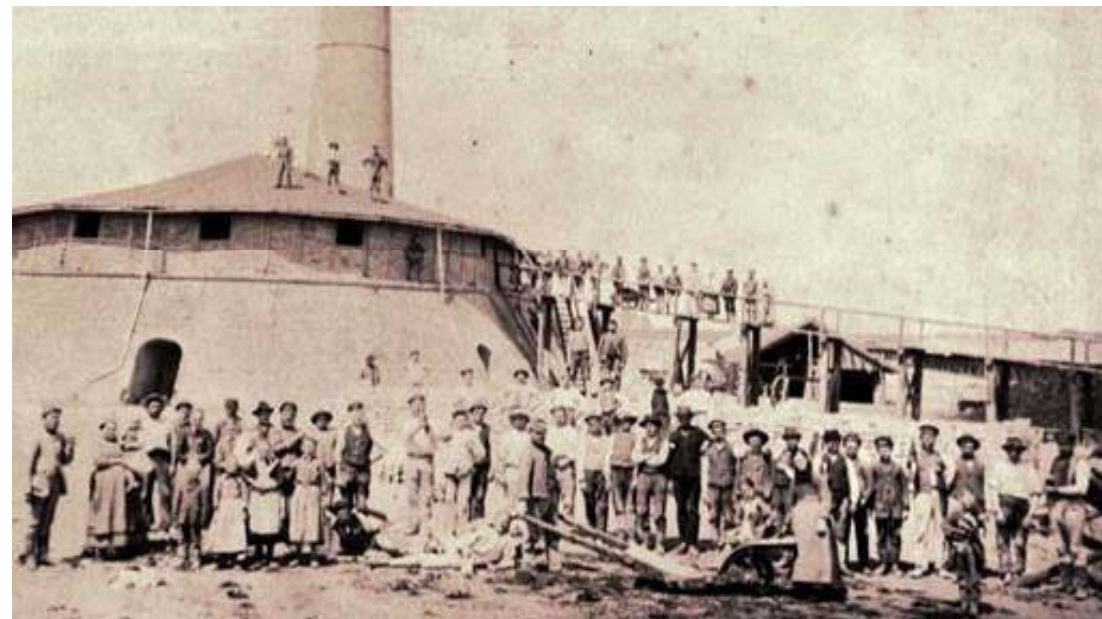
Annotations:

- job config limited to one field by template (points to the 'Configure' link)
- go to template & configure Groovy script generates job config.xml maintain even 10.000s of job configs at once (points to the 'Transformer' section)

How do we use Jenkins?

Challenge #2: Organizing many Projects and Developers

- Use folders and role based access plug-ins to
 - create a project and hierarchy
 - create groups and roles
 - let projects maintain their members themselves
- Live Demo...



How do we use Jenkins?

Folders and role based access plug-ins: Screenshots from Live Demo...

Jenkins
ESI[tronic] 2.0

ESI[tronic] 2.0
Folder name: ESItronic2.0
Projects related to ESI[tronic] 2.0

Releases
Select one of the releases below to see projects related to that release.
To view jobs common to all branches, or development branches, click on the 'Releases' tab.

Releases

All	Common Jobs	Development Branches	Releases
S Name ↓			
Release 2014/1 (8.0) Demo-DVD			
Release 2014/2 (8.1)			
Release 2014/2 (8.1) AA			
Release 2014/2 (8.1) RF4-ORD			

Inherited from ESI[tronic] 2.0

Name
ESI Developers
Project Leaders
ESI Admins

Project Leaders

Roles
manage_groups user

Members

Name
AA-DG/Leo-team-heads

Permissions

Permission	On ESI[tronic] 2.0
Group / View	Granted
Group / Manage	Granted
Role / View	Granted
Job / Read	Granted
Job / Discover	Granted
Job / ExtendedRead	Granted
Job / Build	Granted
Job / Workspace	Granted
Job / WipeOut	Granted
Job / Cancel	Granted
Run / Update	Granted
Run / Artifacts	Granted
View / Read	Granted

Annotations:

- folder config propagates environment to all jobs below it
- add groups and roles to folder inherit to subfolders & jobs
- e.g. Project Leaders assigned to "manage groups" role allows to administrate group members of this project only and update build description, keep builds forever → build admin work delegated to projects!

How do we use Jenkins?

Challenge #3: users often want to create new projects

- Meta-Automation: creating new Folders and Build Projects on demand
- Somewhat similar: let Jenkins monitor Jenkins
- Live Demo...



How do we use Jenkins?

Meta-Automation (1/3), Create Project: Screenshots from Live Demo...

The screenshot shows the Jenkins web interface for creating a new SD system. The breadcrumb trail is: Jenkins > Playground 3 (new folder structure) > Actions > Create new SD system. The left sidebar contains links: Back to Dashboard, Status, Changes, Workspace, Build with Parameters, Delete Project, Configure, Move, Groups, Roles, and Email Template Testing. The main form is titled 'Project Create new SD system' and includes the following fields and labels:

- SWID:** 9978. Label: SD system SWID. Must only be digits (0-9), must be at least 4, at most 10.
- Name:** airbag123. Label: SD system name. Must only contain alphanumeric (a-z and 0-9) characters long.
- Brand:** bmw. Label: Implementation brand name. Must only contain alphabetical characters.
- Group:** (empty). Label: Main group name. Optional.

A 'Build' button is at the bottom of the form. Below the form is a 'Build History' section showing two entries:

- #7 Jun 24, 2014 12:12:01 PM: Created SD system 9978 named airbag123 for brand bmw. SVN URL: ...
- #6 Jun 24, 2014 11:03:48 AM: Created SD system 9978 named ...

creates new build job (from template)
& SVN basic structure

links to newly created
locations build job & svn

The screenshot shows the details for 'Build #7 (Jun 24, 2014 12:12:01 PM)'. The breadcrumb trail is: Jenkins > Playground 3 (new folder structure) > Actions > Create new SD system > #7. The page displays the following information:

- Build #7 (Jun 24, 2014 12:12:01 PM)**
- Created SD system 9978 named airbag123 for brand bmw
- SVN URL: https://rb-svn4.de.bosch.com/svn/t04_Playground3/sd-systems/9978
- SD system job URL: <http://pl-esi-sdbuild.de.bosch.com/job/Playground3-Redux/job/trunk/job/sd-systems/job/9978>
- Specification job URL: <http://pl-esi-sdbuild.de.bosch.com/job/Playground3-Redux/job/trunk/job/specifications/job/9978>
- Revision: 545. No changes.
- Started by user Goettlicher Harald (AA-AS/EIS3-EU)
- This is a snippet.

The screenshot shows the configuration page for 'SD System 9978'. The breadcrumb trail is: Jenkins > Playground 3 (new folder structure) > SD Main Branch trunk > SD Systems > 9978. The left sidebar contains links: Back to Dashboard, Status, Changes, Workspace, Build Now, Delete SD System, Configure, Move, Groups, Roles, and Subversion Polling Log. The main content area displays the following information:

- SD System 9978**
- Full project name: Playground3-Redux/trunk/sd-systems/9978
- This job builds **SD System '9978'** from the *default* development branch **trunk**.
- Most of the job configuration is derived from the template common to all SD System jobs.
- Open this SD System in:
- Web browser: [Raw HTML](#) | [ViewVC](#)
- TortoiseSVN: [Repository Browser](#) | [Log Viewer](#) (requires TortoiseSVN 1.8) | [Checkout](#)
- This job was last regenerated from template on Tue Jun 24 12:12:53 CEST 2014.
- Workspace**
- Recent Changes**
- Build History** (trend)
- [RSS for all](#) | [RSS for failures](#)

Template

This job is created from template "[SD System Component \(3rd Level Template\)](#)"

Continuous Delivery in the Real World

How do we use Jenkins?

Meta-Automation (2/3), Create Branch: Screenshots from Live Demo...

The screenshot shows the Jenkins interface for 'Build #3 (Jun 11, 2014 5:35:31 PM)'. A modal dialog is open with the title 'The page at pl-esi-sdbuild.de.bosch.com says:'. The dialog text reads: 'This will create a branch of SWID 1128 from trunk revision 431936. Please enter the name of the branch you want to create:'. The input field contains 'Jenkins-CDD2014'. There are 'OK' and 'Cancel' buttons. A blue arrow points from the text 'click on link from plus & enter branch name' to the 'Create branch based on this build' link in the left sidebar. Another blue arrow points from the text 'triggers branch creation build' to the 'OK' button.

The screenshot shows the Jenkins interface for the project 'Create a branch of an SD system in Subversion'. The full project name is 'Playground3-Redux/actions/branch-system'. The page includes a 'Workspace' section with 'Recent Changes' and a 'Permalinks' section with a list of build links. A 'Build History' section shows a list of builds, with the most recent one being '#29 Jun 26, 2014 7:27:08 PM' with the description 'Creating SWID 2242 branch Jenkins-CDD2014 ...'.

The screenshot shows the Jenkins interface for the 'SD Branch Jenkins-CDD2014' project. It displays a list of branches: 'SD_Branch dontskipspec', 'SD_Branch foo', 'SD_Branch foobarbaz', 'SD_Branch go82plwashere', 'SD_Branch Jenkins-CDD2014' (highlighted), and 'SD_Branch skipspec'. A blue arrow points from the text 'creates folders, build jobs and filtered/aggregated views' to the list of branches. Another blue arrow points from the text 'contains all build jobs with the same branch name' to the 'SD_Branch Jenkins-CDD2014' entry.

The screenshot shows the Jenkins interface for the 'SD Branch Jenkins-CDD2014' project, displaying a table of build jobs. The table has columns: 'S', 'W', 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The table contains two rows of build jobs.

S	W	Name	Last Success	Last Failure	Last Duration
Blue ball icon	Sun icon	SD Systems > 1128	2 days 7 hr - #1	N/A	51 sec
Blue ball icon	Sun icon	SD Systems > 2242	4 min 38 sec - #1	N/A	1 min 2 sec

Below the table, there are links for 'Icon: S M L', 'Legend', 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'.

How do we use Jenkins?

Meta-Automation (3/3), Jenkins controls Jenkins: Screenshots from Live Demo

- use Groovy System Script to get many infos (with care! test on dev Jenkins!)
- Generate some HTML reports with links to builds, e.g.
 - which build copied which artifacts from where
 - which job ran how often
 - ...has how many script lines
 - ...uses which tools, plugins
- just be creative...!

System Jenkins Jobs

Folder name: System
Jobs for Jenkins statistics, reports and maintenance

[add description](#)

S	W	Name	Last Success	Last Failure	Last Duration	Cre	Trigger
		Backup	N/A	N/A	N/A		
		Diagnostics for 20-16355 (Multiple builds with same ID)	2 hr 46 min - #517	N/A	12 min		Build periodically: H */3 ***
		DiskUsage-Master-JenkinsHome-du.log	1 hr 59 min - #559	N/A	3 m 29 sec		Build periodically: 0 */8 ***
		DiskUsage-Master-JenkinsJobs-du.log	1 hr 56 min - #562	N/A	47 sec		Build periodically: 0 */8 ***
		DiskUsage-pl-eai:Longhorn1_bui9allhome	12 hr - #240	N/A	40 min		Build periodically: 0 5 ***
		JENKINS-CopyArtifactsInfo	1 yr 3 mo - #307	N/A	21 sec		Build periodically: @daily
		JENKINS-JobConfigSearch	9 days 2 hr - #207	1 mo 4 days - #200	1 min 47 sec		
		JENKINS-TriggerLogRotation	3 days 4 hr - #148	N/A	18 sec		
		JENKINS-Update-JenkinsGroovyScripts	1 yr 2 mo - #8	N/A	2 sec		Poll SCM: H */3 ***
		MasterDiskUsage	11 hr - #136	N/A	1.1 sec		Build periodically: 0 6 *** 1-5
		OutOfOrderBuilds	2 mo 26 days - #1	N/A	1 min 16 sec		
		Reports	59 min - #694	N/A	54 sec		Build periodically: 0 ***

Icon: RSS for all RSS for failures RSS for just latest builds

Ant 1.8.4

- BDS - BdsServer-trunk-Java

Gradle

Gradle 1.2

- BDS - BDS-trunk-Java

Gradle 1.3

- Data Purchase - quickreport
- daX - daX-branch-1.0.12-release
- daX - daX-branch-1.0.46-release
- daX - daX-branch-1.0.588-release
- daX - daX-branch-1.0.749-release
- daX - daX-branch-les-2014-1
- daX - daX-DIVAA-trunk_devs_bds
- daX - daX-DIVAA-trunk_devs_download
- daX - daX-trunk-cc-full-nightly/build

About the build if it's stuck

Jenkins build timeout plugin

- KTS 200/340 » Distribution » KTS-Distribution-CardSet
- Leonardo-SD-CSVDATA
- Leonardo-SD-CSVDATA-DiCaFa

[FSTrigger] - Monitor folder

Jenkins FileSystem Trigger Plugin

- EStronic 2.0-all-CROMES-Trigger
- EStronic 2.0-CROMES-all-Trigger
- HP-ALM » HP-ALM-QueueWriter
- KTS 200/340 » SD Production Pool » SDPP-sd/vm394

Publish Javadoc

Javadoc Plugin

- CoVe » Automated_Bundles_Build_Job
- CoVe » CoVe-2.0-Jenkins_Build
- CoVe » daX-CoVe
- CoVe » daX-CoVe-branch-Sprint
- CoVe » daX-CoVe-TRUNK_All_Bundles
- daX » daX-CoVe-branch-275
- daX » daX-trunk-complete
- daX » daX-trunk-full

[Environment Inject] - Setup a build environment

Environment Injector Plugin

- Data Purchase » cove-properties
- DBProductionPool-trunk
- GVCL » GVCL-kits5xx-ng-dev_setup
- GVCL » GVCL-kits5xx-ng_setup
- GVCL » GVCL-kits5xx-ng-fragmented-rpm
- GVCL » GVCL-rpm » kits5xx-ng-rpm
- Leonardo-SD-CSVDATA
- Leonardo-SD-CSVDATA-DiCaFa

[FSTrigger] - Monitor files

Jenkins FileSystem Trigger Plugin

- KTS 200/340 » Database Jobs » DB-2013_2-Trigger
- KTS 200/340 » Database Jobs » DB-2013_3-Trigger

ESI[tronic] 2.0 » Release 2014/2 (8.1) » Database (60597 MB)

- 40 (6023 MB)
- 49 (6023 MB)
- 47 (6023 MB)
- 46 (6022 MB)
- 44 (6085 MB)
- 43 (6085 MB)
- 42 (6085 MB)
- 38 (6084 MB)
- 37 (6084 MB)
- 36 (6083 MB)

ESI[tronic] 2.0 » Release 2014/3 (8.2) » Database (45834 MB)

- 17 (6659 MB)
- 14 (6659 MB)
- 12 (6659 MB)
- 11 (6659 MB)
- 10 (6659 MB)
- 6 (6270 MB)
- 5 (6269 MB)

KTS 200/340 » Database Jobs » KTS200-DB-SD-1ES1-1-trigger has 31 builds

Leonardo » REPORT-LeonardoImplementation-BranchLog has 31 builds

KTS 200/340 » Distribution » KTS 200/340 Distribution to CDNetworks (SD_ERPROBUNG) has 32 builds

KTS 200/340 » KTS-Encryption-SD_ERPROBUNG-Configuration has 32 builds

ESI[tronic] 2.0 » Release 2014/2 (8.1) » UpdateToTestersShare has 33 builds

ESI[tronic] 2.0 » Trunk (9.0) » Setup has 33 builds

KTS 200/340 » Database Jobs » KTS200-DB-SD_TEST has 33 builds

ESI[tronic] 2.0 » Release 2014/3 (8.2) Beissbarth » License Manager has 35 builds

ESI[tronic] 2.0 » Release 2014/2 (8.1) » Update Distribution to CDN has 51 script lines

ESI[tronic] 2.0 » Release 2014/2 (8.1) AA » Update Distribution to CDN has 51 script lines

ESI[tronic] 2.0 » Release 2014/2 (8.1) BOOT (HUB) » Update Distribution to CDN has 51 script lines

ESI[tronic] 2.0 » Release 2014/2 (8.1) BOOT-POC » Update Distribution to CDN has 51 script lines

ESI[tronic] 2.0 » Release 2014/3 (8.2) » Update Distribution to CDN has 51 script lines

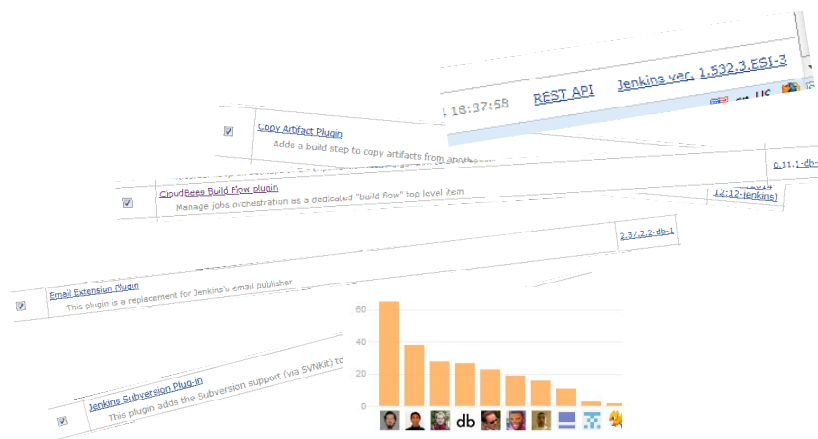
ESI[tronic] 2.0 » Release 2014/3 (8.2) Beissbarth » Update Distribution to CDN has 51 script lines

ESI[tronic] 2.0 » Release 2015/1 (9.0) Beissbarth » Update Distribution to CDN has 51 script lines

How do we use Jenkins?

Challenge #4: going to the limits brings up many (special) bugs

- Partially patched releases / plug-ins needed
- Bug fixing is possible with Open Source
 - Build up in-house know-how
- Professional Support helps
 - e.g. Jenkins Enterprise



The Cloudbees team helping to deliver our bricks...

Conclusion

- Continuous Delivery allows instant results even for challenging size and complexity
- Jenkins can be used for such challenging projects
 - Many plug-ins and customizations needed
 - Enterprise plug-ins can be helpful
- Professional support can be helpful
- Still deep in-house know-how should be in place



Göltzschtalbrücke 1846-1851
world's largest brick bridge
574 x 78m – 26 M bricks

Thank you!

For questions

- harald.goettlicher@de.bosch.com
- https://www.xing.com/profile/Harald_Goettlicher

Kudos to Daniel Beck for development

- danielbeck@beckweb.net

And remember:
Don't let your software
end up like this –
do continuous delivery!

