Esdk Best Practice Workshop

Oder auch: wie arbeiten wir ohne eine grüne Spielwiese im Alltag?

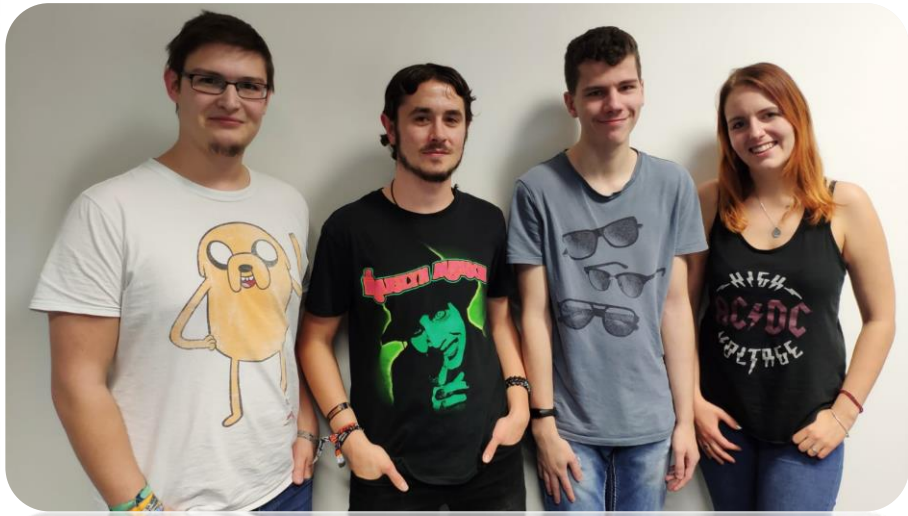
abas ERP

WIR STELLEN UNS VOR

ABAS GMBH: TEAM PRODUCT

- Bestehend seit 01.01.2019
- Know How in den Bereichen Web-Entwicklung und ERP
- Produkte von uns:
  - abas Mobile Warehouse (Web basiert)
  - (neues) abas Mobile Shopfloor (Web basiert)
  - abas Connect (Web basiert)
  - QM-App (ERP / esdk-App)
  - AEB-App (ERP / esdk-App)
  - Weitere folgen

Steven Wruk David Bannasch



Patrick Huber Jasmin Marquardt

QM-APP: DIE ERFAHRUNGEN

Agenda

TAG 1

- esdk Projekteinrichtung
- Git
- Issuemanagement
- Docker
- Praxis

TAG 2

- Resources Ordner
- Build.gradle
- Datenbanken
- Masken
- Daten
- Aufzählungen
- Infosysteme
- Praxis

TAG 3

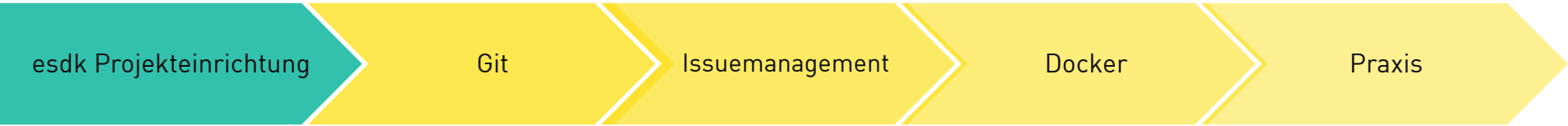
- CleanCode
- Helper Klassen
- EventHandler Templates
- Unit-Tests
- Integration Tests
- Praxis

ZUSATZ

SONSTIGE WICHTIGE THEMEN

Agenda

TAG 1



TAG 2

TAG 3

ESDK PROJEKTEINRICHTUNG

Schritt 1 – esdk Projekt generieren lassen

-> https://dev.abas-essentials-sdk.com/#/project-builder

-> https://documentation.abas.cloud/en/esdk-project-initializer/

Schritt 2 – Für ein neues Projekt neuste abas-tools verwenden

-> https://extranet.abas.de/sub_de/download-bereich/erp/apis-tools/abas-tools-download.php

Schritt 3 – initGradleProperties.sh ausführen (in Windows Power Shell *./initGradleProperties.sh*)

Schritt 4 – Docker container hochfahren (in Windows Power Shell *docker-compose up -d*)

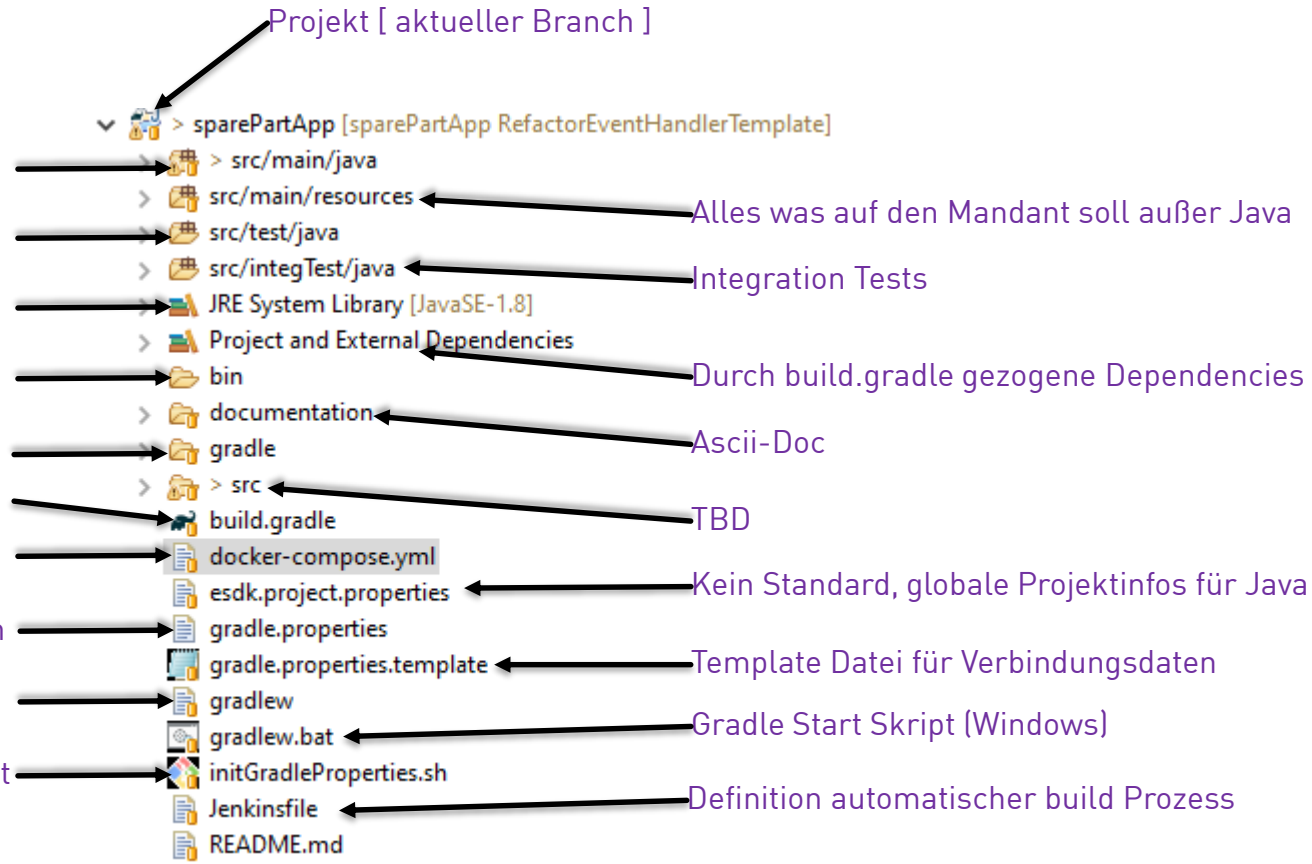
Schritt 5 – Projekt als gradle Projekt importieren (Bei Bedarf installierte gradle Version auswählen!)

Schritt 6 – Beispielklassen löschen oder in separaten Ordner verschieben

- Ich habe Dummerweise bei meinem Punkt „Resources Ordner“ das gesamte Projekt gemacht… Folie hab ich dir hier mal reingepackt (mit fancy Animation) evtl willst du sie ja verwenden

WIEDERHOLUNG

ESDK PROJEKT



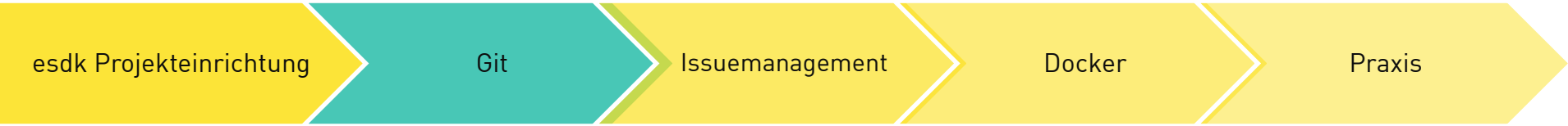
ESDK PROJEKTEINRICHTUNG

Anmerkungen

- Nicht ein altes Projekt kopieren, immer neu anlegen. Esdk entwickelt hier stetig weiter wodurch immer wieder hilfreiche Neuerungen wie z.B. Beispiel Programme Tests etc reinkommen
- Bei abasTools Nutzung: Projektordner esdk und workspaceOrdner separat halten, vermeidet unnötigen overload durch workspace Daten im Projekt und man spart sich das ganze in die gitignore rein zu nehmen
- Wenn es nach der Anlage von deinem Error bei dem ersten fullinstall o.ä. zu Fehlern kommt (vorher prüfen) aus der build.gradle NEXUS_USER und NEXUS_VERSION löschen, hat bei mir geholfen
- Gradle.properties: Nexus_Host darf keinen Hosts Eintrag enthalten (muss IP oder Notebook name sein), Rest darf über eine Hosts geregelt werden wenn gewünscht
- Gradle.Properties: oft kommt es bei Tasks zu verschiedenen Fehlermeldungen der Art: keine ssh Konfig möglich, Fehler in build.gradle line XX in Verbindung mit Host erwähnt. Wenn es geklappt hat und von heut auf morgen solche Fehler auftreten: bei Nexus_Host statt z.B. notebookname mal IP probieren etc hilft meist.

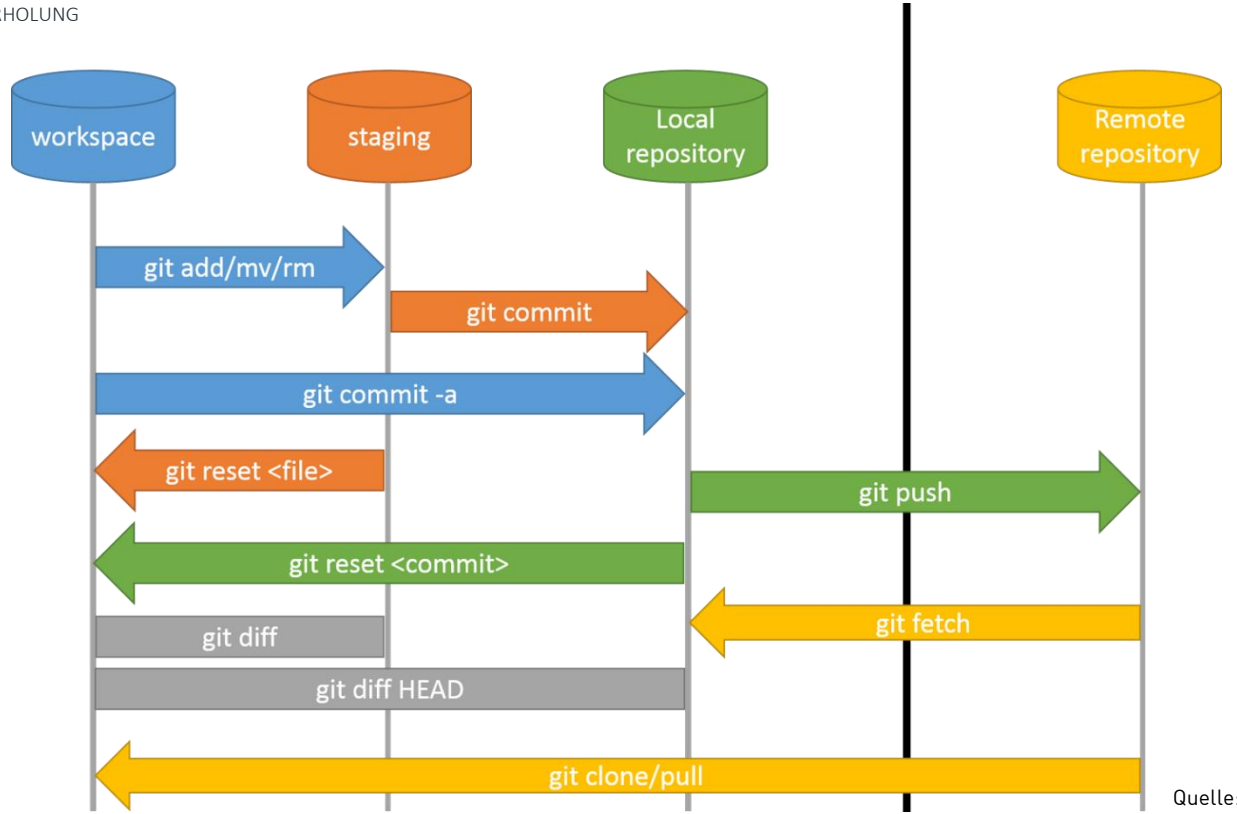
Agenda

TAG 1



TAG 2

TAG 3



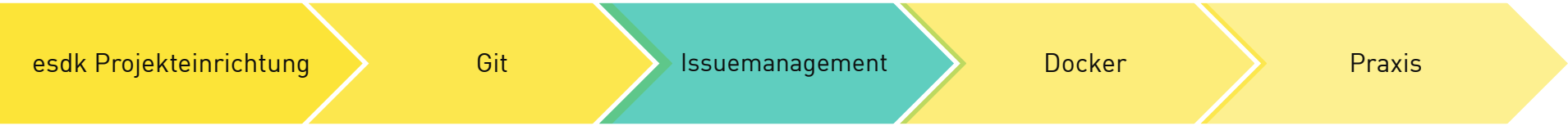
Quelle:
https://twiki.cern.ch/twiki/bin/view/BL4S/BL4SGitGuide

GIT

- Jeden Commit überprüfen
- Kleine, thematisch zusammengehörige Commits
- .gitignore initial pflegen (erspart viel Arbeit)

Agenda

TAG 1



TAG 2

TAG 3

WIEDERHOLUNG

ISSUEMANAGEMENT

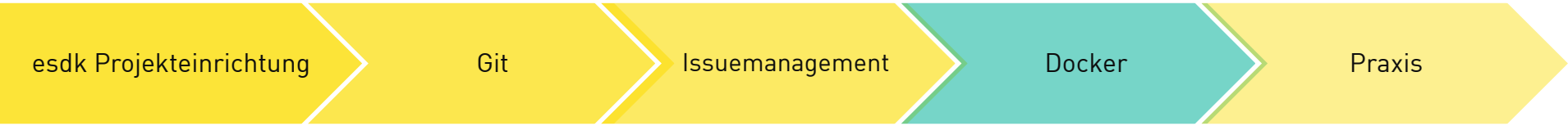
- Empfohlen: Zentrales Tool für alle
- Ziel: Gemeinsame Übersicht über Projektstand
- → Live Demo

ISSUEMANAGEMENT

- Disziplin sind A & O, sonst Tool unnötig
- Einigung auf gemeinsame Terminologie
- Nicht jeder sollte Issues anlegen (Filterung durch PO & Team)
- Nutzen der verfügbaren Felder (Version, Milestone, Verfasser)
- Features auch in abas PM nutzbar (Kanban Board)

Agenda

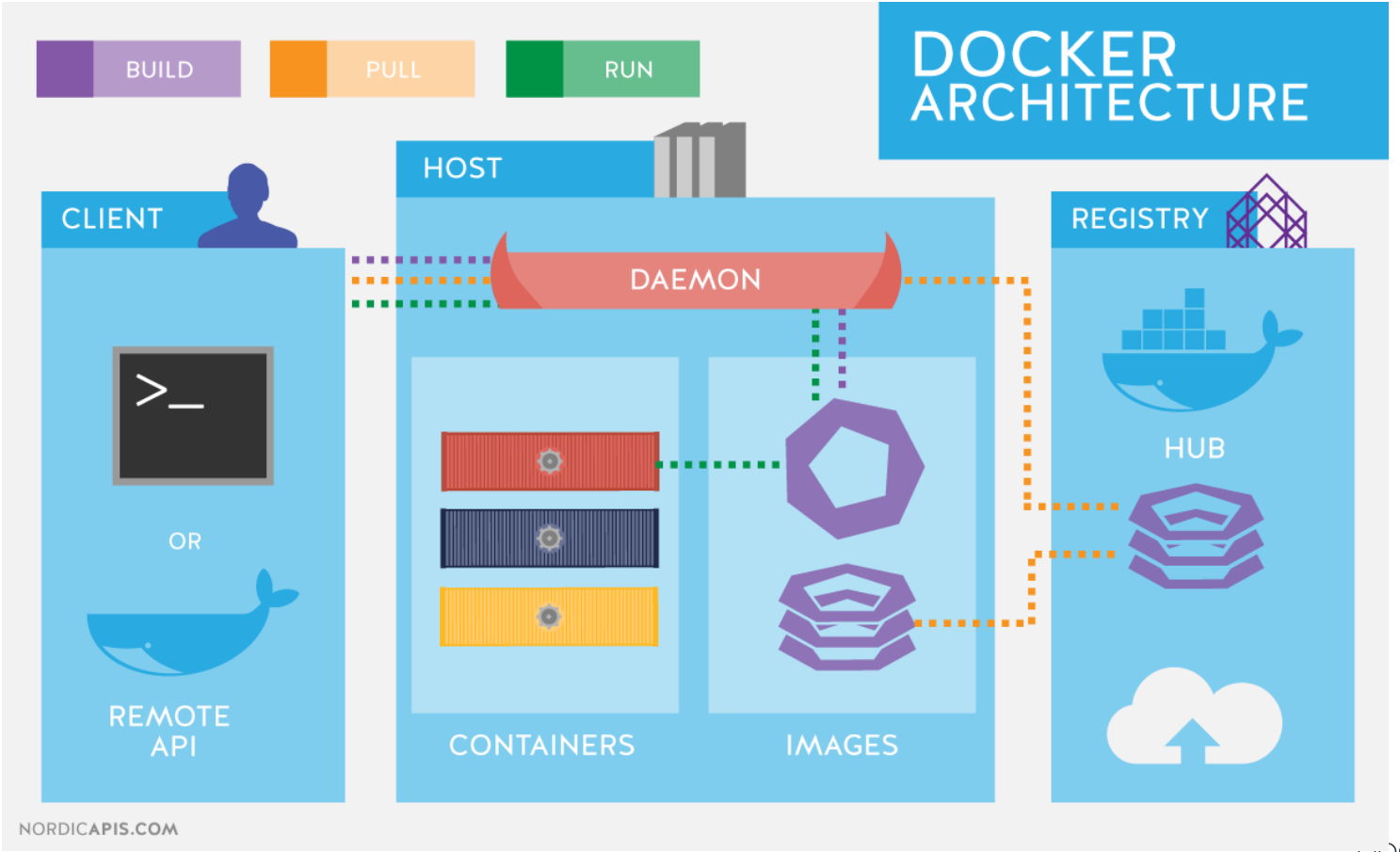
TAG 1



TAG 2

TAG 3

DOCKER


BUILD
PULL
RUN
DOCKER ARCHITECTURE
HOST
CLIENT
DAEMON
REGISTRY
HUB
OR
REMOTE API
CONTAINERS
IMAGES
NORDICAPIS.COM

DOCKER-COMPOSE

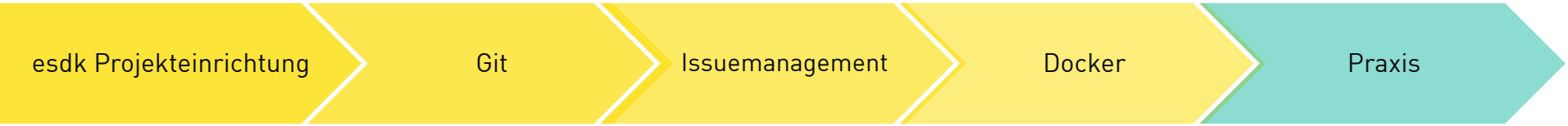
```yaml
version: '3.7'
services:
    # The ERP Container
    # -----------------
    # Accessing container via ssh: docker exec -u erp -it erp bash
    #
    erp:
      image: partner.registry.abas.sh/abas/test:2018r4nl3
      init: true
      container_name: "erp"

      ports:
      - "${SSH_TEST_PORT:-2205}:22"
      - "${MINI_GUI_PORT:-8001}:80"
      - "${EDP_TEST_PORT:-6560}:6550"
      - "${GUI_TEST_PORT:-48592}:48392"

      environment:
      - ABAS_HOST=${ABAS_HOST}
      - ABAS_GUI_PORT=${GUI_TEST_PORT:-48592}

      # The hostname is very important, especially if you intend to use `docker commit`
      user: s3
      hostname: dockerbau
      command: ["sh", "-c", "cd /abas/erp && eval $$(sh denv.sh) && datmeta -s && /data/starteVersion.sh run" ]

    # nexus for publishing abas Essentials libraries
    nexus:
      image: sonatype/nexus:oss
      container_name: "nexus"
      ports:
      - "8081:8081"
```

DOCKER

- Connection-Testing aus Docker heraus (Stichwort: gradle.properties, docker exec)
- Achtung bei Integration Tests: Eigene Daten nutzen
- Troubleshooting: Docker for Windows: Docker selbst + Docker Service neu starten
- Eigene Images mit docker commit erstellen (Spart Zeit, wenn zeitaufwendige Aktionen durchgeführt wurden)
- Achtung Lizensierung bei abas Docker Container laufen ab

Agenda

TAG 1

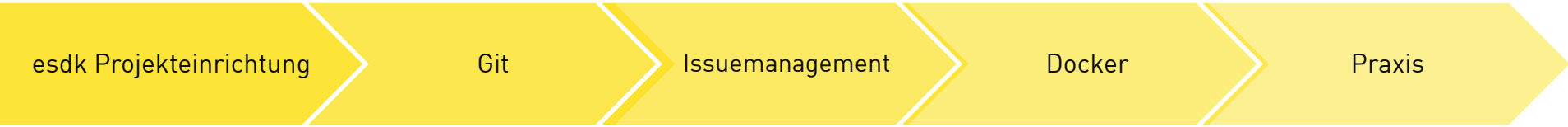


TAG 2

TAG 3

Agenda

TAG 1



TAG 2

TAG 3

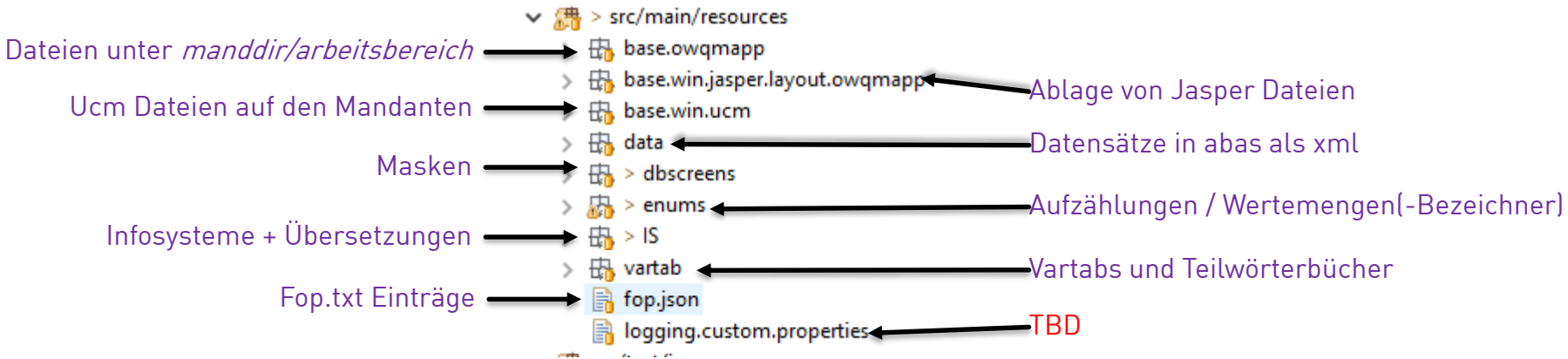
Agenda

TAG 1

TAG 2

TAG 3

DER RESOURCES ORDNER



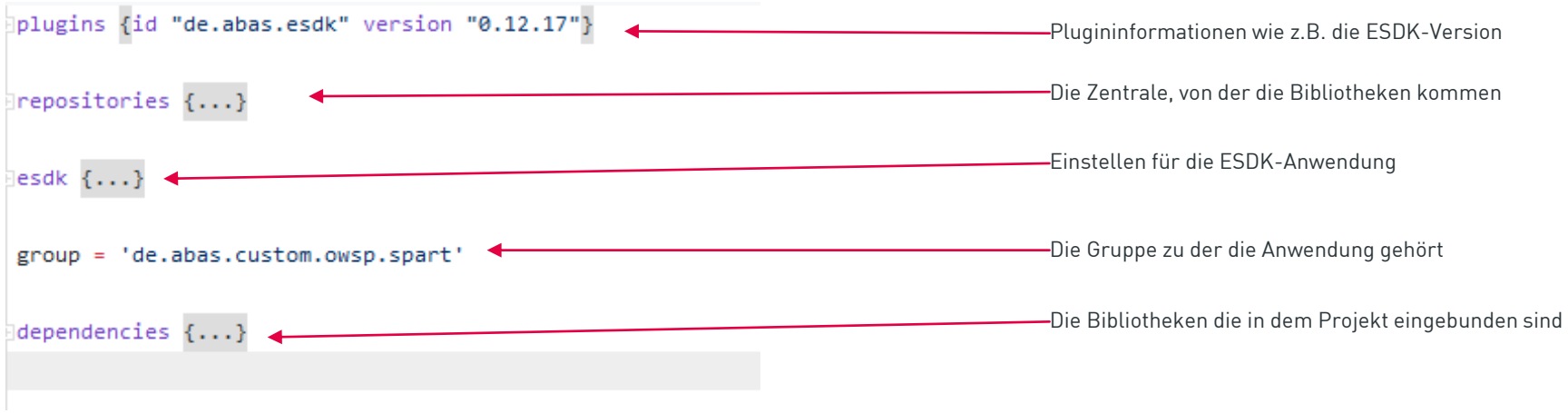
Agenda

TAG 1

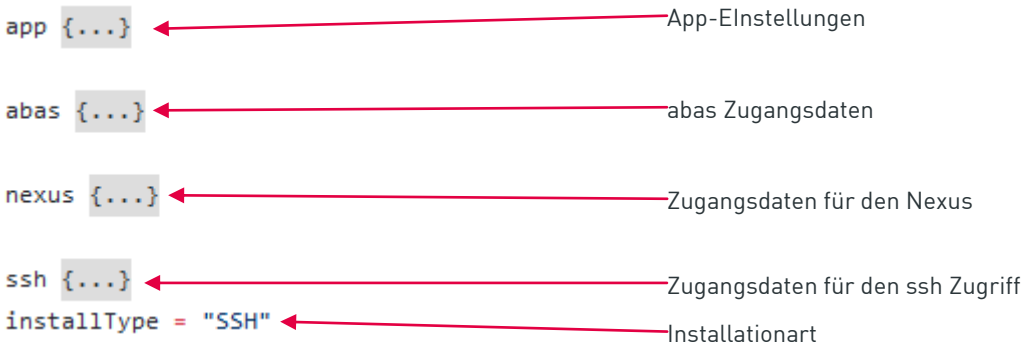
TAG 2

TAG 3

BUILD.GRADLE



BUILD.GRADLE



BUILD.GRADLE

```
app {
    name = "sparePartApp"
    vendorId = "es"
    appId = "spart"
    shared = true
    infosystems = ["IS.OWSPART.SPUREASON"]
    tables = ["ErsatzteileApp", "Teil"]
    screens = ["ErsatzteileApp:Konfiguration":["A"], "ErsatzteileApp:Ersatzteile":["A"], "2":["A","D"]]
    data = []
    enums = ["UsageReason"]
    workdirs = ["owspart"]
    namedTypes = []
    languages = "D"
}
```

Masken, welche hinzugefügt werden

Enums, die hinzugefügt werden

Mandantenverzeichnisse

BUILD.GRADLE

```
ssh {
   host = SSH_HOST
   port = SSH_PORT.toInteger()
   user = SSH_USER
   password = SSH_PASSWORD
   key = SSH_KEY
   timeout = 25000
}
```

Agenda

TAG 1

TAG 2

TAG 3

Vartab im Mandant erweitern

Variablenname: yAPPIDvarname

In build.gradle hinzufügen

tables = ["ErsatzteileApp", "Teil"]

ExportAll oder ExportVartab

ExportAll oder ExportVartab

Teilwörterbücher löschen sofern nicht benötigt

Aktuell läuft bei install noch bei jedem Teilwörterbuchung Übersetzung inkl. Maskengenerierung → ewige Laufzeit

Leere Gruppen aus StandardVartabs löschen

Es werden immer alle Gruppen, auch ohne neue Variable exportiert. Dies kann (z.B. bei EK/VK) zu Fehlern bei installVartab führen

Exportierte Datenbanken umbennen – führende Nummer

Die Zusatzdatenbank beginnt mit 01_ , der Rest wird fortlaufend benannt. Sofern bei dem nächsten Export keine Variable geändert/hinzugefügt wurde können die exportierten .schm (ohne Nummer) gelöscht werden (dadurch weniger Aufwand Gruppen löschen). Ausnahme: ist in der Vartab eine, durch die App generierte Aufzählung als Variablenart hinterlegt muss immer die neu exportierte Datei genommen werden

Agenda

TAG 1

TAG 2

TAG 3

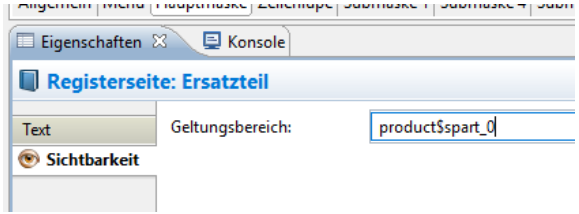
MASKEN

Schritt 1 – Masken individualisieren

Schritt 2 – Masken bearbeiten (extra Tab)

Mittlerweile können Masken um beliebig viele Tabs erweitert und beliebig benannt werden

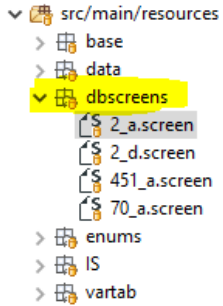
-> In Sichtbarkeitstext des Tabs *<Engl. Datenbankbez.>$<appid>_<tabnr.>* eintragen



Schritt 3 – Masken in build.gradle eintragen (Maskennummer/Datenbanknummer und Priorität)

```
shared = true
infosystems = ["IS.OWSPART.SPUREASON"]
tables = ["ErsatzteileApp", "Teil"]
screens = ["ErsatzteileApp:Konfiguration":["A"], "ErsatzteileApp:Ersatzteile":["A"], "2":["A","D"]]
data = []
enums = ["UsageReason"]
```

Schritt 4 – gradle task „exportScreens“ ausführen



MASKEN

- Bei AbasTool 1.4.1 (alias 1.5) KEINE Masken im Projekt bearbeiten (sondern ScreenEditor), führt im Projekt bei install zu irreführenden Fehlern (z.B. Taks prepareVartab bricht mit Fehler „null“ ab oder Aufzählung kann nicht mehr importiert werden mit Fehlermeldung „Klassenname doppelt“ > in abasTools bearbeitete Maske löschen und es geht wieder). Scheinbar ab neueren abasTools Versionen möglich

Agenda

TAG 1

TAG 2

TAG 3

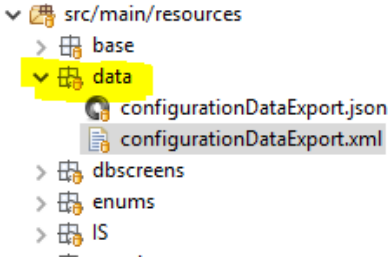
DATEN

Schritt 1 – Daten im Docker-Mandanten anlegen

Schritt 2 – json Datei definieren (Datensätze die exportiert werden sollen)

-> Vorher herausfinden, was auf jeden Fall gebraucht wird -> Pflichtfelder

Schritt 3 – Gradle Task „exportData“ ausführen -> XML-Datei wird generiert



JSON: Was brauche ich?

```
{
  "roots" : [
    {
      "dbnr" : 15,
      "grnr" : 0,
      "headfields" : [ "id", "guid", "such", "name" ],
      "tablefields" : [],
      "criteria" : "id=(153,15,0)"
    }
  ]
}
```

XML: Ergebnis

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ABASData>
    <RecordSet tableNumber="15:0" tableName="ErsatzteileApp:Konfiguration" identifier="guid" standard="false"
        <Record id="(153,15,0)">
            <Head>
                <Field name="id" abasType="ID15">(153,15,0)</Field>
                <Field name="guid" abasType="GLA38">273282ef-90a3-468c-83fe-27b772fa44c2</Field>
                <Field name="such" abasType="SWK15">KONFIG</Field>
                <Field name="name" abasType="T34">Konfiguration für ErsatzteileApp</Field>
            </Head>
        </Record>
    </RecordSet>
</ABASData>
```

DATEN

- Daten nach erfolgreichem Export wieder aus build.gralde entfernen / json in Projekt liegen lassen falls nochmal export notwendig
- Nummernkreise und Druckinfrastruktur über Nachbrenner
- Aufrufparameter z.B. 59000 -> Zeilen aktuell nicht additiv. Möglich das dies gerade in Umsetzung ist.
- In der json Datei erstmal definieren, was man unbedingt an Daten braucht. Esdk teilt einem eventuelle Abhängigkeiten mit und generiert auch gleich das json Schema. So spart man sich Arbeit.

Agenda

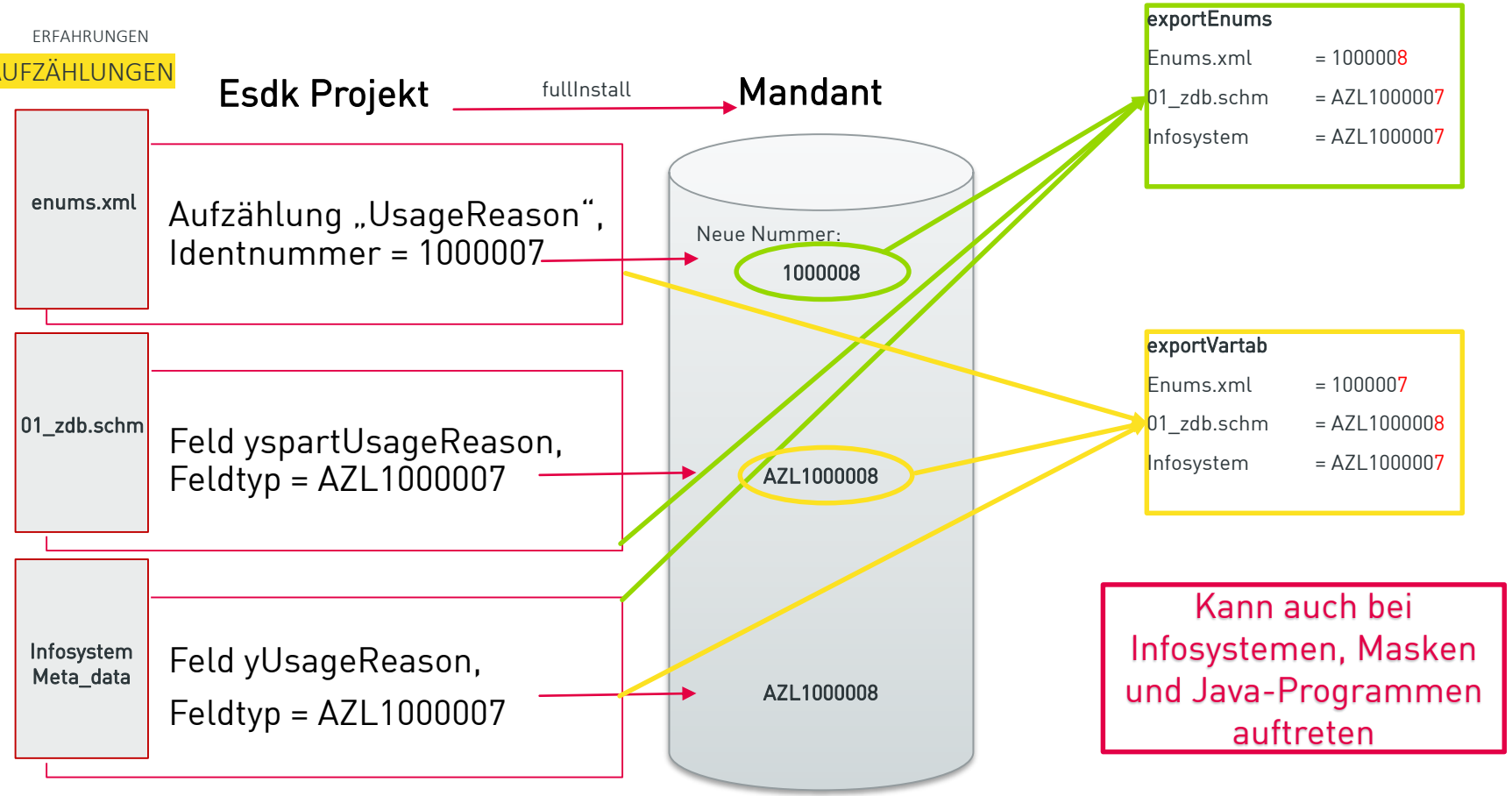
TAG 1

TAG 2

TAG 3

AUFZÄHLUNGEN

ACHTUNG: Falsche Handhabung kann das Projekt zerschießen



ACHTUNG: Falsche Handhabung kann das Projekt z

ERFAHRUNGEN

AUFZÄHLUNGEN

Kann auch bei Infosystemen, Masken und Java-Programmen auftreten

abas ERP

ACHTUNG: Falsche Handhabung kann das Projekt zerschießen

- Führt auch zu großen Problemen bei arbeiten mit Git im Team
- Sehr umfangreiche Mergekonflikte
- Manchmal nicht als Mergekonflikt sondern als geänderte Zeilen angezeigt

Gegenmaßnahmen:

- Änderungen der Aufzählungsnummern immer in separaten Branch
- Master in den aktuellen Branch mergen und erst schauen ob fullInstall noch läuft und Masken noch in Ordnung sind
- Wenn exportAll nur wegen eines Datensatz ohne Aufzählungsnummer ausgeführt wird (z.B. Datenbank ohne AZL) → temporär alles andere aus der build.gradle entfernen

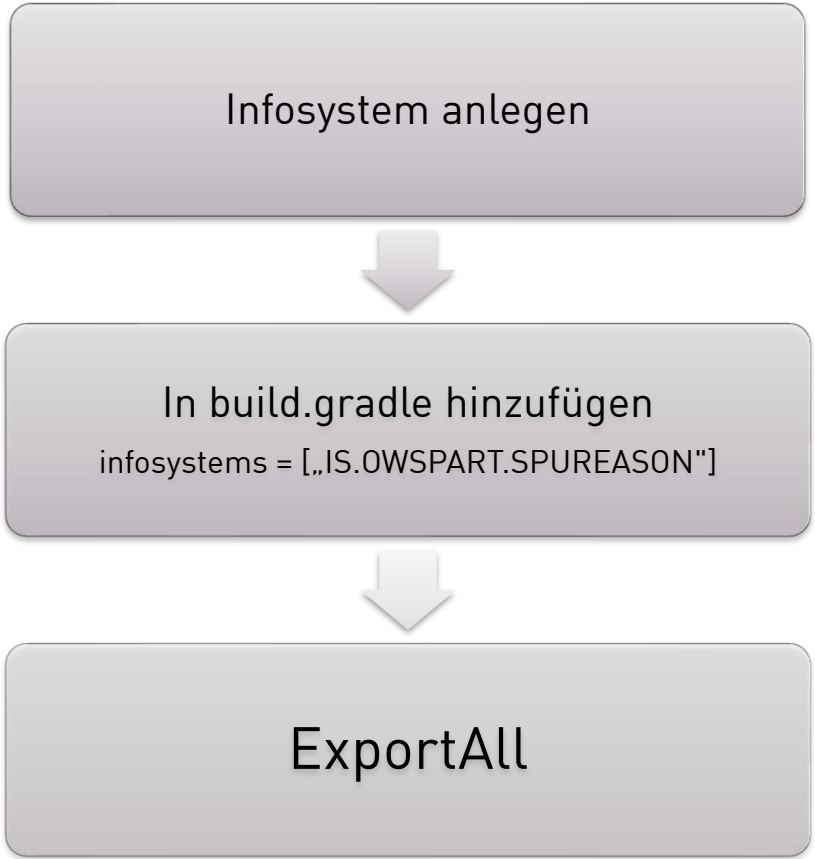
Agenda

TAG 1

TAG 2

TAG 3

INFOSYSTEME



INFOSYSTEME

Agenda

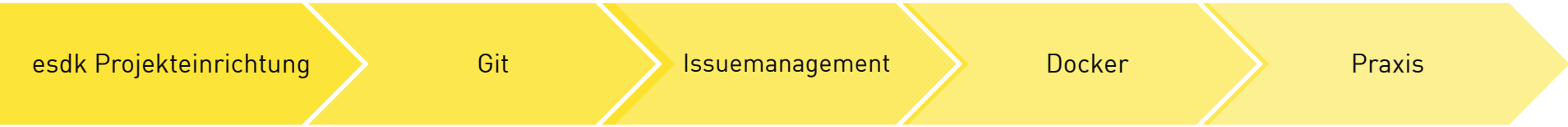
TAG 1

TAG 2

TAG 3

Agenda

TAG 1



TAG 2

TAG 3

Agenda

TAG 1

TAG 2

TAG 3

CLEANCODE

Static Code Metrics	Cyclomatic Complexity	Must be less than 21.
	Maximum Method Length	Max LOC/functions must be less than 31.
Dynamic Code Metrics	Tests	Test types are not restricted. You are free to use Unit Tests, Integration Tests and/or End-to-End Tests. All tests must pass.
	Test Coverage	Test Coverage must be greater than or equal to 80%.
	Test Execution	Test execution cannot exceed 1 hour.

CLEAN CODE

- Don‘t Repeat Yourself
- Single Responsible Princible
- Naming
- Self documenting code

- Ziel:
  - Weniger Wartungsaufwand
  - Leichtere Testbarkeit

Agenda

TAG 1

TAG 2

TAG 3

HELPER KLASSEN

- Helper-Klassen sind ausgelagerter und oft benutzter Code
- Dienen dazu sich nicht ständig zu wiederholen (DRY -> don´t repeat yourself)
- Der Code wird sauberer und besser lesbar
- Wiederverwendbarer Code

HELPER KLASSEN

VORHER

```
public void screenEnter() throws EventException {
   if (EnumEditorAction.New.equals(event.getCommand()) || EnumEditorAction.Copy.equals(event.getCommand()) ) {
      if (!systemInformation.getEdpMode()) {
         throw new EventException("nicht erlaubt", 1);
      }
   }
}
```

HELPER KLASSEN

NACHHER

```
public void screenEnter() throws EventException {
   if (codeTemplates.isEventNewOrCopy(event) ) {
      codeTemplates.actionIsProhibitedInGui();
   }
}
```

HELPER KLASSEN

- CodeTemplates
- EsdkProperties
- SystemInformation
- CustomizationAccess
- DatabaseMetaData
- FileContentReplacer
- RuntimeLicenseChecker
- UCMFileProcessor

Agenda

TAG 1

TAG 2

TAG 3

EVENTHANDLER TEMPLATE

Hier einleiten kurz „Problemstellung“:

- Vermeiden von redundantem Code > alle Infosysteme haben immer den gleichen Aufbau:
- > Check License bei ScreenEnter und / oder ButtonStart
- > vor und nach jedem Event Aufruf von individualisierung

Dadurch nicht nur unschöner Redundanter Code, sondern bei Änderungen auch Nacharbeiten an extrem vielen Stellen (Beispiel QM-App: ca 120 Aufrufe der FopMethode vorher, jetzt nur noch 6)

Agenda

TAG 1

TAG 2

TAG 3

UNIT TESTS

UNIT TESTS

Agenda

TAG 1

TAG 2

TAG 3

WIEDERHOLUNG

INTEGRATION TESTS

- Lib der esdk Abteilung zeigen, dass mann damit nicht nur Datenbankverbindung herstellen kann sondern auch einfach Errors/Messages vom Mandant mitbekommt, schnell Daten selektieren und löschen kann etc → https://documentation.abas.cloud/en/esdk-javadoc/test-utils/
- Nochmal kurz grundlegend sagen was Integration Test ist

INTEGRATION TESTS

- Anfangs sehr zäh reinzukommen, aber sehr hilfreich da man recht schnell mitbekommt wenn durch eine Änderung was kaputt geht
- Bei qmapp erst nachträglich eingefügt -> gleich bei dem ersten Test einen relativ schlimmenBug gefunden
- Anlegen von Testdaten sollte gut aufgegliedert, in verschiedene Objekte etc… gepackt sein, da sonst bei einem Fehler der Testdatenerzeugung alle Tests in denen die jeweilige Datenerzeugung aufgerufen wird mit nicht sprechenden Fehlermeldungen fehlschlagen → sehr schwierige Fehlersuche

Agenda

TAG 1

TAG 2

TAG 3

ZUSATZ

SONSTIGE WICHTIGE THEMEN

ZUSATZ

SONSTIGE WICHTIGE THEMEN

WIEDERHOLUNG

RELEASE EINER APP

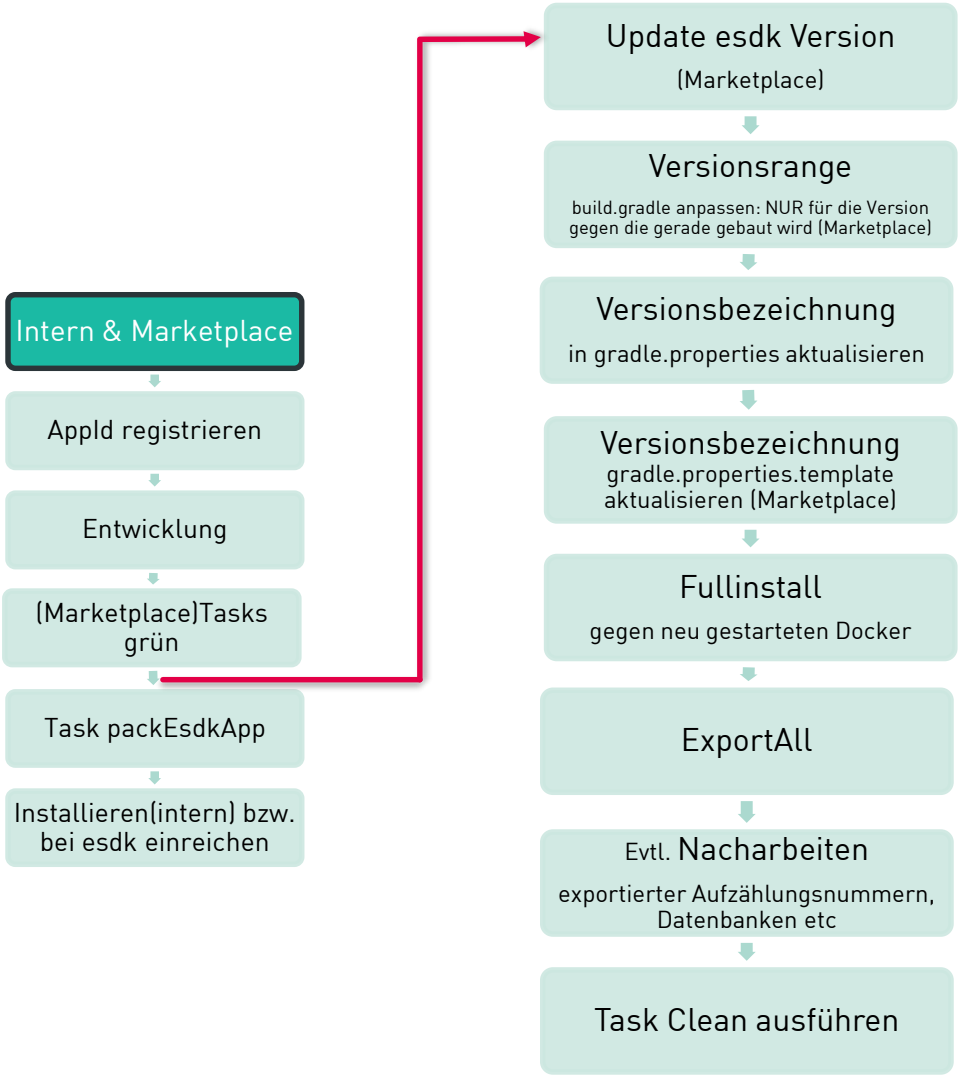
App für interne Verwendung

- AppId registrieren
- Entwicklung
- Task packEsdkApp
- Standalone AppJar aus dem zip kann installiert werden

Marketplace App

- AppId registrieren
- Entwicklung
- Tasks calculateCodeCoverage und CodeCoverageVerification müssen grün durchlaufen
- packEsdkApp
  - Je abas Version eine AppJar bauen (2016/2017/2018...)
- Jars einzeln im esdk Portal zur Registrierung einreichen

RELEASE EINER APP



ZUSATZ

SONSTIGE WICHTIGE THEMEN

LIZENZIERUNG

- In build.gradle muss unter dependencies licensing "de.abas.esdk:client-api:0.0.3:all„ eingetragen werden
- Was du noch rausfinden müsstest: muss das bei esdk oder sonst wo bekannt gegeben werden? // Kamil mal anschreiben
- Die Lizenz wird an zwei Stellen geprüft 1. bei installation mit AppInstaller 2. zur Runtime wenn Methode aufgerufen wird

```
public static void validateLicense() throws FOPException, EventException {
boolean isValidLicense = LicenseChecker.instance().validate();
if (!isValidLicense) {
throw new EventException(EXIT_ERROR_CODE); }}
```

- Auf dem Kundenserver muss CloudConnect laufen damit die Kommunikation zum AG Lizenzserver stattfinden kann

LIZENZIERUNG

- Schwer damit gegen einen Docker zu testen (der hat kein CloudConnect, ist nicht lizenziert etc)
- ➢ Wir hatten daher bisher die Lizenzabfrage zur Entwicklung auskommentiert, ist aber nicht schön. Richtiger Weg: Docker bei Ag lizenzieren lassen und ein Image mit CloudConnect, Internetverbindung etc vorbereiten
- Ansonsten Recht unkompliziert, Fehler kamen bisher eigentlich nur bei Installation hoch und waren dann immer Sache der Technik (Proxy Einstellungen, CloudConnectDocker nicht gestartet etc.)
- Wenn ein Kunde bei dem alles lief auf einmal Fehlermeldung „Invalid License“ bekommen hat > hat bisher immer daran gelegen, dass der Server neu gestartet wurde aber CloudConnect nicht im Auto Start war

ZUSATZ

SONSTIGE WICHTIGE THEMEN

EXTERNE LIBARY EINBINDEN

-> auch sagen, dass danach refresh gradle project ausgeführt werden muss

EXTERNE LIBARY EINBINDEN

Zusatz

SONSTIGE WICHTIGE THEMEN

ASCII DOC

ASCII DOC

Hilfreiche Links

WO KANN MAN DENN WAS NACHSCHAUEN?

Esdk – Dokumentation:

https://documentation.abas.cloud/en/esdk/

Refactoring:

https://refactoring.guru/refactoring

Docker Befehle:

https://docs.docker.com/engine/reference/commandline/docker/

Musterlösung der Übungsaufgabe:

https://github.com/JayJay692/web-dev-gmbh

VIELEN DANK!

abas ERP