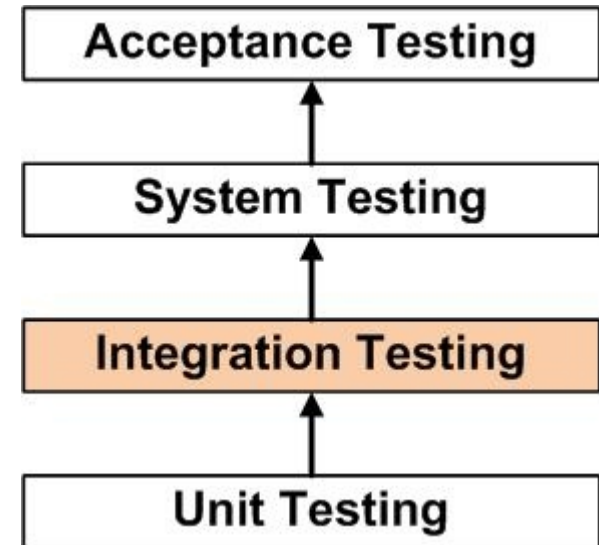


Dry-Run Procedures & Sequencer Tutorial

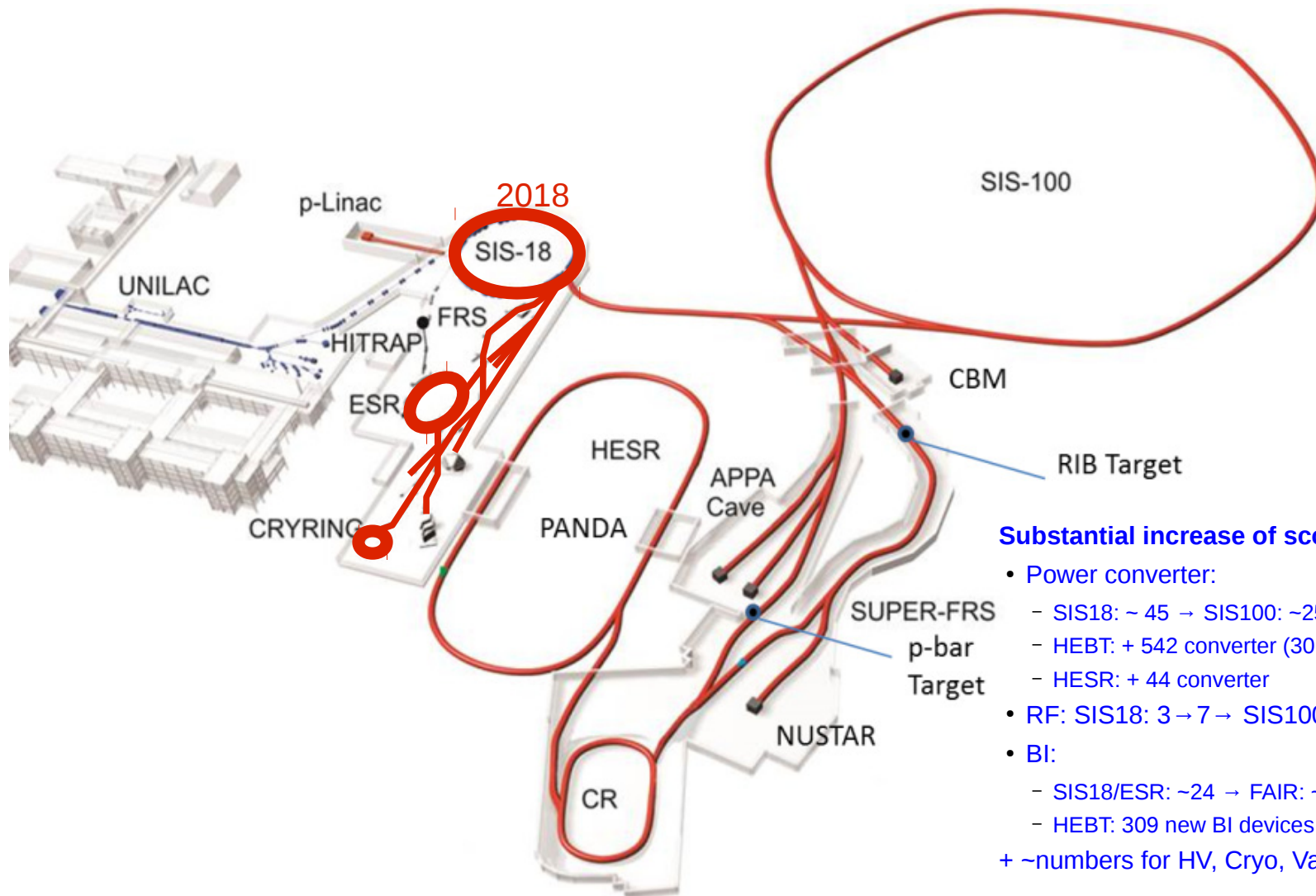
Ralph J. Steinhagen

Accelerator Operation



Equipment

image courtesy: <http://softwaretestingfundamentals.com/>



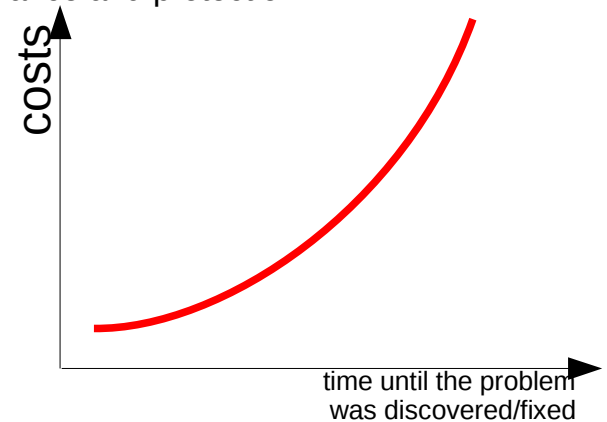
Substantial increase of scope w.r.t. GSI:

- Power converter:
 - SIS18: ~ 45 → SIS100: ~250 converter
 - HEFT: + 542 converter (30 types!)
 - HESR: + 44 converter
 - RF: SIS18: 3 → 7 → SIS100: 38 RF sys.
 - BI:
 - SIS18/ESR: ~24 → FAIR: ~250 BPMs
 - HEFT: 309 new BI devices
- + ~numbers for HV, Cryo, Vacuum, TI, ...

... minimise common mistakes, procedural errors, etc. affecting machine performance and protection.

→ Fix problems early, when and where they occur:

- Minimises procrastination of errors
- “Safety starts with safe habits”!



• Benefits of early indicators of developing/not-yet-critical faults:

– Identification/isolation of faults during HWC/DryRuns (e.g. via Sequencer)

fixes “domino effect” problems at the source not its symptoms:

- HWC: parallel testing possible → resource control by SPL/equipment group
- BC: predominantly sequential testing → limited/no resource control ↔ “fire-fighting”

– Preventative maintenance

- early indication of drifting parameters (e.g. slow thermal runaway of IGBTs)
- group common/similar maintenance tasks rather planning for having react (minimises “fire fighting”)

– Post-Mortem analysis ('as good as new' SIL assurance)

- insures safety-integrity-level, minimises risk at the source before they escalate
- minimises human-intervention – OP/expert has to handle errors that have not/cannot be automatically analysed by the post-mortem system

- Procedures verify that functional and ‘Detailed Specification’ requirements have been fulfilled → FAT, SAT but also importantly for Beam Operation
 - checks MoU between various stake-holders (AP, BI, CO, RF, ...)
 - *define when, where and how the individual accelerator systems should fit in*
 - identify loop-holes/missing critical functionality
 - identification/isolation of errors & regression testing
 - *ie. minimises that errors fixed in the past are not being reintroduced*
 - ensures that HW is largely/fully commissioned prior to BC/operation
 - *N.B. HWC (testing in ||) vs. BC (testing predominantly sequentially)*
 - (semi-)automation: better to ‘act’ than to ‘react’ → triangle of ‘work to be done’ vs. ‘required project timeline’ vs. ‘available resources’
 - traceability and documentation (for your peers, SPL/management, operation, ...)
 - improve the quality of the project (QA)
- Test procedures describe the intention of what needs to be achieved
 - specifically not implementation or tool specific ie. implementation can be refactored without the need to change the intention of the test
 - terminology/hierarchy of: unit testing → integration testing → acceptance testing

- Description
 - test procedures describe the intention of what needs to be achieved
 - specifically not implementation or tool specific
- Entry Conditions
 - dependencies to other systems/conditions → reference to other test sequences
- Machine Setup
 - conditions to be checked beforehand
- Procedure – list of individual tasks
 - for each detailed testing step:
 - short-hand key for each step (e.g. 'step A', naming taken from DS document)
 - short description of what needs to be achieved (ie. a 'one liner')
 - acceptance criteria (either from DS or split into warning/error)
- Problems
 - be free to comment on (technical) concerns or potential issues
- Exit conditions
- Open Questions & Action Items
- References & Acronyms
 - where applicable, helps you with further QA documentation

The screenshot shows a web browser displaying the FAIR Commissioning & Control Working Group (FC2WG) website. The page title is "BeamCommissioning - BeamCommissioning - FC2WG - FAIR Wiki - Google Chrome". The URL is <https://fair-wiki.gsi.de/foswiki/bin/view/FC2WG/BeamCommissioning/WebHome>. The page features a navigation menu with tabs: Home, Minutes, Next Agenda, Dry-Runs, Commissioning Procedures, Control Topics, and Admin. The main content area is titled "You are here: FC2WG » BeamCommissioning" and contains a timeline diagram of the commissioning process. The timeline is divided into three main phases: Hardware Commissioning (6 months), Commissioning with Beam (3-6 months), and Assisted Operation. The Commissioning with Beam phase is further divided into Stage A (Pilot Beams), Stage B (Intensity Ramp-up & Special Systems), and Stage C (Production Operation with nominal Intensities). A red box highlights the "Stage A - Pilot Beams" section, which is noted as the "Main focus for 2018 (re-commissioning, new CO)".

Stage A - Pilot Beams

Main focus for 2018 (re-commissioning, new CO)

- **main aim:**
 - drive the beam expeditiously through the [BeamProductionChain](#) (BPC): from the sources, through the synchrotrons, beam transfers, up to the experimental targets/storage rings
 - check basic 'accelerator mechanics': threading, injection, capture, cool, convert, acceleration/decelerate, stripping & extraction
 - identify beam-related limitations: polarities, RF, beam instrumentation, machine alignment, effective physical machine aperture, ...
- always done with 'safe' resp. low-intensity/brightness beam
 - initially with 'easily available' ions (e.g. U28+, Ar -> simpler optics, beam dynamics, etc.), then protons (tests transition crossing, etc.)

Stage B - Intensity Ramp-up & Special Systems

- **main aim:**
 - achieving and maintaining nominal machine performance for a limited number of reference beam
 - check that the accelerator design and systems can achieve (near) nominal beam parameters, e.g. beam intensities, nominal transmission and beam losses for e.g. U28+ & proton beams, etc.
- commissioning of e.g. e-cooler (if not needed earlier), slow extraction, transverse fast feedbacks
- commissioning and validation of machine protection & interlock systems
- possibly unsafe operations always preceded by checks with safe beam

Stage C - Production Operation with nominal Intensities

- **main aim:**
 - make fast setup and switch-over between different [BeamProductionChains](#) routine
 - push physics and beam parameter performance (intensity, brightness/emittance, momentum spread, ...)
 - identify and improve upon bottlenecks impacting FAIR's figure-of-merit
 - improve the machine model using beam-based techniques

FAIR Commissioning Phase A.1) - Injection and First Turn initial test in 2018

Last modified by [Wolfgang Geithner](#) on 07. Mar 2018 - 09:29 - r11

- Description
- Entry Conditions
- Machine Setup
- Procedure
 - Details of activities
- Problems
- Exit conditions
- Open Questions & Action Items
- References
- Acronyms

Description

- Commissioning of the last section of the preceding transfer line (matching section + few metres before) and the injection region
- First commissioning of key beam instrumentation
- Commissioning of the trajectory acquisition and correction
- Threading the ring (first turn)
- Closing the orbit to be ready for phase [A.2 Circulating Pilot Beam](#)

Entry Conditions

[Show...](#)

Machine Setup

[Show...](#)

Procedure

Step	Activity	Who	Priority	Special Procedures YR GS GE 1S CR HR
A.1.1	Commission Injection Region (1 Pilot Bunch)			
.01	Commission final metres of preceding TL		1	X
.02	Setup injection elements with beam		1	
.03	Beam commissioning injection screens & grids		1	
.04	Detailed steering onto moved-in injection collimator (if available -> otherwise vacuum chamber)		1	
.05	Power injection kicker/bumper		1	
.06	Check stability of transfer lines and trajectory without injection kicker/bumper		2	
.07	Check stability of injected trajectory with injection kicker/bumper ON, measure kicker/bumper waveform		2	
.08	Beam commissioning Software-Interlock-System (parasitic)		2	
.09	Perform aperture scan at magnetic/electro-static injection septa		2	
A.1.2	Threading Beam around the Ring			
.01	Open injection absorber/collimator (if applicable, or not dumped onto vacuum chamber)		1	
.02	Coarse beam commissioning of BPMs (asynch. acquisition/narrow band, if available)		1	
.03	Commission trajectory acquisition and correction		1	
.04	Threading around ring		1	
.05	First measurement of energy mismatch (correction if needed)		1	
.06	First BPM and corrector polarity checks and repairs		1	
.07	Beam commissioning of DCCTs & ICT		1	
.08	Commission BPM intensity measurement mode (parasitic)		2	

CommandModeTest - HardwareCommissioning - FC2WG - FAIR Wiki - Google Chrome

FW CommandModeTest - x

Secure | <https://fair-wiki.gsi.de/foswiki/bin/view/FC2WG/HardwareCommissioning/CommandModeTest>

Apps GSI CERN - LHC CERN - TEMP Misc Optics 3D Printing RF-Components Kickstarter RF Jobs CERN Frankfurt Elektro Other bookmarks

FAIR Commissioning & Control Working Group (FC²WG) search

Home Minutes Next Agenda **Dry-Runs** Commissioning Procedures Control Topics Admin

HWC Overview Sequencer **Equipment Tests** Controls EPC BI HV RF other Dry-Run

You are here: [FC2WG](#) » [HardwareCommissioning](#) » [ControlSystemProcedures](#) » [PowerConverterProcedures](#) » [GenericConverterStatusTest](#) » [CommandModeTest](#) Edit Attach New More

EPC CommandModeTest Description

Last modified by [RalphSteinhagen](#) on 07 Dec 2017 - 10:09 - r3

Test basic Command Mode functionality (for the time being only 'DC pulsed mode')

- check interfaces
- switch state 'on' and 'off'
- do simple low-level device trims + actual vs. reference comparison checks

Some variable definition (to be shifted to LSA/device specific variables)

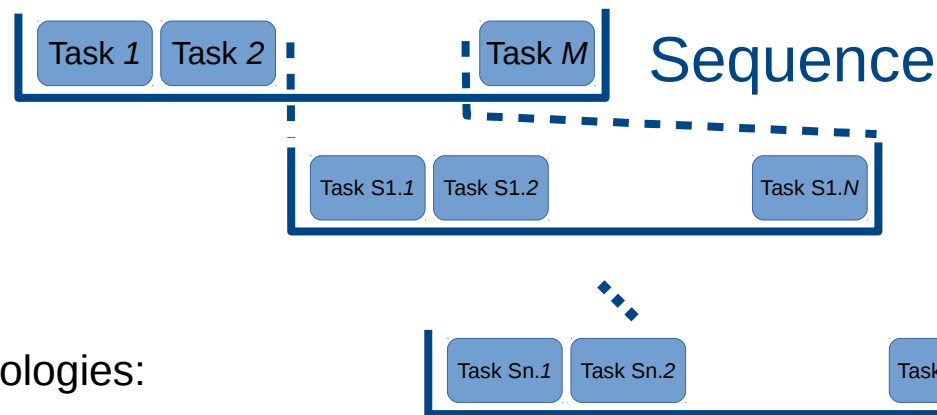
- I_MIN: PC minimum current as defined in FESA 'Setting' property
- I_MAX: PC maximum current as defined in FESA 'Setting' property
- currentStep = DEFAULT_CURRENT_STEP = 10 A -> 10% of PC I_MAX
- currentTolerance = Max(DEFAULT_CURRENT_TOLERANCE = 1 A, 1% of I_MAX)
- DEFAULT_SLEEP = 5 s

Procedure

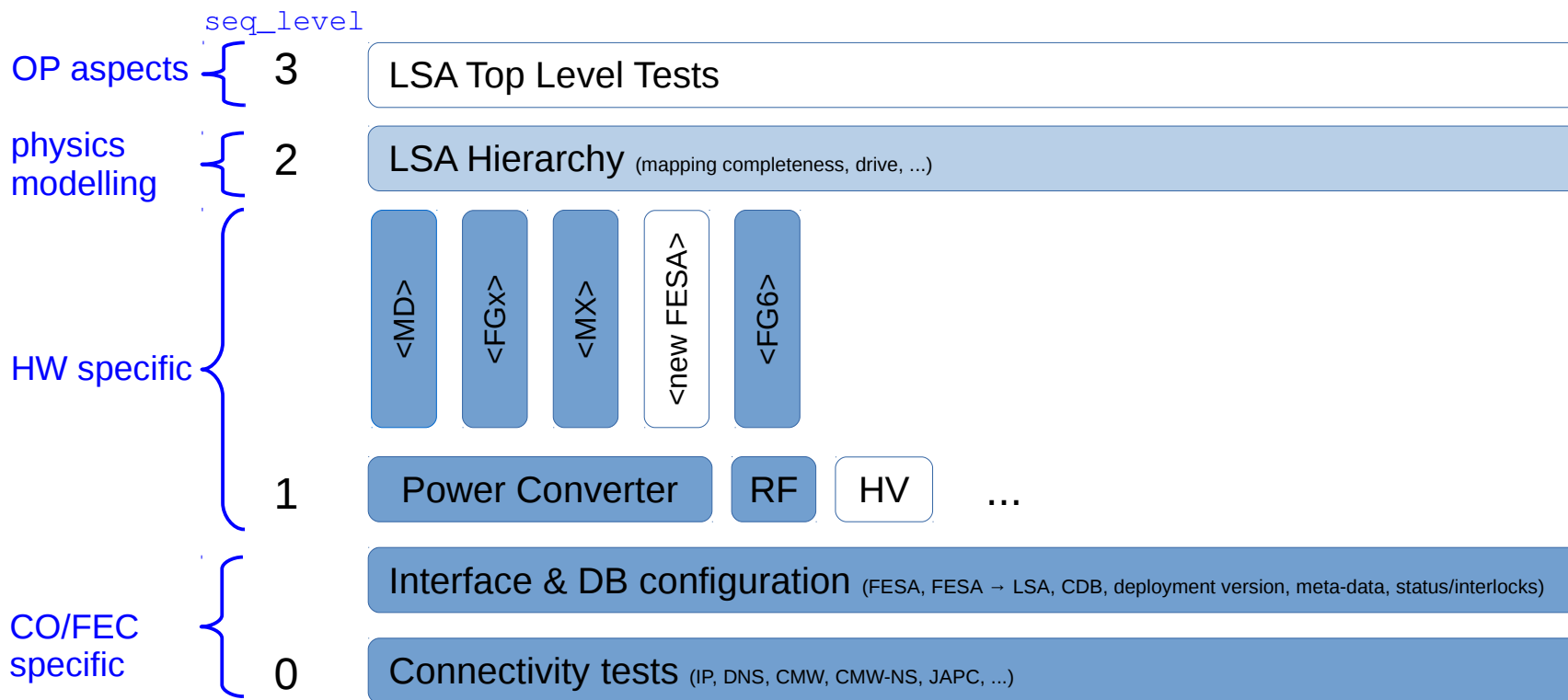
Step	S?	short description	Criteria
prepare		init device status handler	
step1	Y	FrontendVersionTest	
step2	Y	GenericConverterStatusTest	
step3		check for required FESA interfaces	
step4		switch converter to DC mode 'true'	
step5	Y	read I_min/I_max, check their consistency, set current step and tolerance	
step6	Y	check initial value is zero	I_actual < currentTolerance
step7	Y	set DC-value to 'currentStep'	
step8	Y	wait for DEFAULT_SLEEP seconds	
step9	Y	check 'actual-reference' value for non-zero current	I_actual-currentStep < currentTolerance
step10	Y	set DC-value to zero	
step11	Y	wait for DEFAULT_SLEEP seconds	
step12	Y	check 'actual-reference' value for zero current	I_actual < currentTolerance
step13	Y	check that reference setting writes with DC mode 'false' are invalid	
step14	Y	check that writes on actual value property 'DCValueAcq' are invalid	
testStep		switch converter to DC mode 'false'	

Hello [Ralph Steinhagen](#)
 Log out
 Create personal sidebar
Toolbox
 Create New Topic
 Index
 Search
 Changes
 Notifications
 RSS Feed
 Statistics
 Preferences
Webs
 FC2WG
 BeamCommissioning
 HardwareCommissioning
 Main
 Sandbox
 System

- ‘Task’ = device class specific atomic test, e.g.
 - connectivity test, power ‘on’, power ‘off’, ...
 - actual vs. reference comparison, ...
- ‘Tasks’ can be assembled to ‘Sequences’ ...
...which may also contain further sub-sequences:



- CO backbone technologies:
 - FAIR Archiving Systems → Documentation
 - LSA-based Settings Management → Reference & Data Supply
 - System- and Site-wide Digitisation of Analog Signals → ‘actual vs. reference’ process monitoring



- Some logstash meta-data keys (see: <https://logstash.acc.gsi.de/>):
 - Existing tags: program: 'sequencer', user_name, pid, ...
 - seq_device: e.g. device name, LSA property name, global function
 - seq_level: <0 ... 4>, seq_task: <task/class name>, seq_sequence: <collection of tasks> (???)
 - seq_testID: unique identifier for given sequencer run (↔ multi-user, parallelism)
 - seq_test_start: <time-stamp>

- What is provided by the sequencer frame-work:

```
abstract class AbstractSequenceImpl extends AbstractTaskImpl implements Sequence {
    void exec() {
        execPreUserCode(); // init communication to Archiving, LSA, Sequencer Service etc.
        execUserCode();
        execPostUserCode(); // reporting, etc.
    }
}
```

- Level 1 tests (provided by the CO/equip. Group/machine experts):

```
class FrontEndGenericConnectivityTest extends AbstractSequenceImpl implements Sequence {
    // [...]
    void execUserCode() {
        addTask(new Task("prepare", "init device status handler", this, () -> {
            deviceConnection = new GenericStatusTest(deviceName);
        }));

        addTask(new SkippableTask("step1", "check simple IP connectivity", this, () -> {
            if (deviceConnection.isHostReachable(CONNECTIVITY_TEST_IP) == false) {
                throw new RawSequenceException("testCMW3get() - failed");
            }
        }));
    }
    // [...]
}
```

```
class HwcTest1 extends AbstractSequenceImpl implements Sequence {
    void execUserCode() {
        addTask(new FrontEndGenericConnectivityTest(getParameter()));

        addTask(new Task("step1", "description #1", seqReference, () -> {
            task1(); // user/device-specific non-skippable atomic test operation 1
        }));

        addTask(new SkippableTask("step2", "description #2", seqReference, () -> {
            task2(); // user/device-specific skippable atomic test operation 2
        }));
        [...]
    }

    void task1() {
        // test SAT-A sub-procedure x.1, see specification... item ...
        // [...], may throw Exception (or derivatives)
    }

    void task2() {
        // test SAT-A sub-procedure x.2, see specification... item ...
        // [...], may throw Exception (or derivatives)
    }
}
```

Some examples to get a flavour of the targeted code style and flavour:

<https://www-acc.gsi.de/svn/applications/app-codesnippets/>

<https://www-acc.gsi.de/svn/applications/sequencer/>

default - Discover - Kibana - Mozilla Firefox

https://logstash.acc.gsi.de/kibana/app/kibana#/discover/default?indexPattern=logstash-*&type=histogram&a=(columns:[logsource,level,Properties.seq_device,Properties.seq_task,message],filters:[{"state":{"store":appSt...

Discover Visualize Dashboard Settings

user_name: "rstein" Actions

logstash*

Selected Fields

- level
- logsource
- message
- Properties.seq_device
- Properties.seq_task

Available Fields

Analyzed

Indexed

Type

Field name

Hide Missing Fields

Reset Filters

Popular

- @timestamp
- _index
- _source
- _type

Sequencer Highlight All Match Case Whole Words

18.Apr.2017, 13:27:27.815 - 18.Apr.2017, 13:42:27.815 - [by 30 seconds](#)

Count

@timestamp per 30 seconds

Time	logsource	level	Properties.seq_device	Properties.seq_task	message
18.Apr.2017, 13:41:06.356	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XYRT1QD62' available as deployed version '0.4.2', interlock = false, opReady = true
18.Apr.2017, 13:41:06.356	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	total of 104 devices available
18.Apr.2017, 13:41:06.356	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	total of 27 devices unavailable
18.Apr.2017, 13:41:06.355	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XYRT1INIX' available as deployed version '0.4.2', interlock = false, opReady = true
18.Apr.2017, 13:41:06.355	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XYR03KV7K' available as deployed version '0.4.2', interlock = false, opReady = true
18.Apr.2017, 13:41:06.355	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XYR03M03C' available as deployed version '0.4.2', interlock = false, opReady = false
18.Apr.2017, 13:41:06.355	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XPLPSIZ1C1' available as deployed version '0.4.2', interlock = false, opReady = false
18.Apr.2017, 13:41:06.355	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XYR03KV6G' available as deployed version '0.4.2', interlock = false, opReady = true
18.Apr.2017, 13:41:06.354	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XYRT1QD61' available as deployed version '0.4.2', interlock = false, opReady = true
18.Apr.2017, 13:41:06.354	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XYR03KV3G' available as deployed version '0.4.2', interlock = false, opReady = true
18.Apr.2017, 13:41:06.354	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XYR03KV4K' available as deployed version '0.4.2', interlock = false, opReady = true
18.Apr.2017, 13:41:06.354	asl743.acc.gsi.de	INFO	Sequencer Summary	Sequencer Main Class	device 'XYR03KH3G' available as deployed version '0.4.2', interlock = false, opReady = true

DryRunII TestSequence Report (LENIENT)

description: basic Dry-Run II checks in view of Beam Commissioning in 2018
 Test Start: 2017-11-15 07:40:46, Test ID: -2350101035315585321

Tested Devices:

- **SIS18:** GS00BEF, GS01QS1F, GS01QS2D, GS02BE1F, GS08BE2F, GS11MU2, GS12QS1F, GS12QS2D, GS12QS3T
- **GSI_HEBT:** GTE1KY1, GTE1QD11, GTE1QD12, GTE2KX1, GTE2QT11, GTE2QT12, GTE2QT13, GTE3MU1, GTH1KX, GTH1QD11, GTH1QD12, GTH2KY1, GTH2QD11, GTH2QD12, GTH2QD21, GTH2QD22, GTH3KY1, GTH3QD11, GTH3QD12, GTH4KS1, GTH4KY1, GTH4MU1, GTH4MU2, GTH4QD11, GTH4QD12, GTH4QD21, GTH4QD32, GTS1MU1, GTS7MU1, GTV1MU1, GTV2MU2, GTV2MU3, GTV2QD11, GTV2QD12.
- **Other:** 2018 Dry-Run II Test, GHHTMU1, GHTBMU1, GHHTKY1, GHHTQD11, GHHTQD12.

Tested Sequences:

Sequence Name	OK	Skipped	FAILED
BasicLSATest	1	0	0
CommandModeTest	45	0	0
DryRunII_TestSequence	53	0	0
FecConnectivityTest	52	0	0
FecStandardTest	52	0	0
FecStatusTest	1	0	0
FecVersionTest	7	0	0
GenericConverterStatusTest	23	0	0
MiniMaspTest	1	0	0

Detailed Test Sequences:

Sequence:Task Name:	detailed description	OK	Skipped	FAILED
DryRunII_TestSequence:DryRunII_TestSequence	basic Dry-Run II checks in view of Beam Commissioning in 2018	1	0	0
MiniMaspTest:prepare	init device status handler	1	0	0
BasicLSATest:prepare	init device status handler	1	0	0
DryRunII_TestSequence:prepare	prepare global dry-run sequence	1	0	0
DryRunII_TestSequence:BasicLSATest	checks basic LSA functionalities and interfaces	1	0	0
DryRunII_TestSequence:MiniMaspTest	checks basic MASP functionalities and interfaces	1	0	0
MiniMaspTest:check1	test availability and consistency of all_status property	1	0	0
BasicLSATest:check1	check if 'SIS18' 'B' ho' is defined and retrievable	1	0	0
BasicLSATest:check2	check if 'CRYRING' 'B' ho' is defined and retrievable	1	0	0
MiniMaspTest:check2	test consistency of MASP POWER_ON with device status	0	1	0
MiniMaspTest:check3	test consistency of MASP INTERLCK with device status	0	1	0
MiniMaspTest:check4	test consistency of MASP MOD_RDY with device status	0	1	0

Sequence:Task Name:	detailed description	Result:
- step3	check availability/conformity deployed versions	FINISHED
- step4	check LSA vs deployed versions	FINISHED
FecStandardTest:FecStatusTest	simple FESA status property test	FINISHED
FecStatusTest:prepare	init device status handler	FINISHED
- check1	test availability/consistency of FESA Status property	FINISHED
- check2	report on pending device warning and error messages	FINISHED
- check3	check whether device is in DC mode	FINISHED
- check4	check whether Status->Status is 'OK'	FINISHED
- check5	check device power state	SKIPPED_FAULTY
- check6	check device being in remote state	SKIPPED_FAULTY
- check7	check device being in (SW) interlock	FINISHED
- check8	check device being in OPready mode	FINISHED
- check9	check whether device modules are ready	SKIPPED_FAULTY

GS12QS2D:

Sequence:Task Name:	detailed description	Result:
DryRunII_TestSequence:FecStandardTest	standard FEC test: connectivity -> version vs. LSA -> status	FINISHED_FAULTY
FecStandardTest:FecConnectivityTest	low-level front-end communication and status tests	FINISHED_FAULTY
FecConnectivityTest:prepare	init device status handler	FINISHED
- step1	check simple IP connectivity	FINISHED
- step2	check availability of RDA nameserver	FINISHED
- step3	check low-level CMW3 'get'	FINISHED_FAULTY

GS12QS3T:



- N.B. no nice GUI (yet) but report similar to previous...

Simple Sequencer GUI Test <@as1741.acc.gsi.de>

Available Sequences:

Filter:

- RecoverDryRunI_TestSequence
- sequences.SuperFRS
- MagneticPrecycle
- PrecycleAllFrsMagnets
- sequences.controls

Loaded Sequences:

Filter:

TaskName	Device	Result
DemoSequence(F)	DemoDevice	FINISHED_FAULTY
unknownTaskName	DemoDevice	FINISHED
taskName1	DemoDevice	FINISHED
taskName2	DemoDevice	FINISHED
taskName3(S)	DemoDevice	SKIPPED
taskName4(F)	DemoDevice	FINISHED_FAULTY
taskName5 (5/5)	DemoDevice	NOT_STARTED
taskName6	DemoDevice	NOT_STARTED
shortTaskName (0/5)	DemoDevice	NOT_STARTED
DemoNestedSequence	DemoDevice	NOT_STARTED
DemoSequenceUnit	DemoDevice	NOT_STARTED
DryRunIII_TestSequence	DryRunIII_T...	NOT_STARTED
prepare	DryRunIII_T...	NOT_STARTED
BasicLSATest	DryRunIII_T...	NOT_STARTED
MiniMaspTest	DryRunIII_T...	NOT_STARTED
FecStandardTest	GS00BEF	NOT_STARTED
FecStandardTest	GS12QS2D	NOT_STARTED
FecStandardTest	GS12QS3T	NOT_STARTED
FecStandardTest	GS11MU2	NOT_STARTED
FecStandardTest	GS01QS1F	NOT_STARTED
FecStandardTest	GS01QS2D	NOT_STARTED
FecStandardTest	GS12QS1F	NOT_STARTED
FecStandardTest	GS02BE1F	NOT_STARTED
FecStandardTest	GS08BE2F	NOT_STARTED

execute more execute buttons (stop, cancel, step, ...)

Selected Sequence:

B	S	Task	Description	Status	P *
		DemoSequence	short description of what sequence is supposed to do	FINISHED_FAULTY	
		unknownTaskName	unknownTaskDescription	FINISHED	
		taskName1	taskDescription 1	FINISHED	
	S	taskName2	taskDescription 2 -- this test can be skipped	FINISHED	
	S	taskName3	taskDescription 3 -- this test can and is skipped by default	SKIPPED	
		taskName4	taskDescription 4 -- this will fail but cannot be recovered	FINISHED_FAULTY	
	S	taskName5	taskDescription 5 -- this test can be skipped	NOT_STARTED	
	S	taskName6	yet another task	NOT_STARTED	
X	S	shortTaskName	???	NOT_STARTED	

Detailed Task Status

Task Name: taskName4
Device: DemoDevice
Task Description: taskDescription 4 -- this will fail but cannot be recovered
Task Status: FINISHED_FAULTY
Task Parent: DemoSequence
Break Point set?: YES
Skippable?: NO
Skip Set?: NO
Documentation:

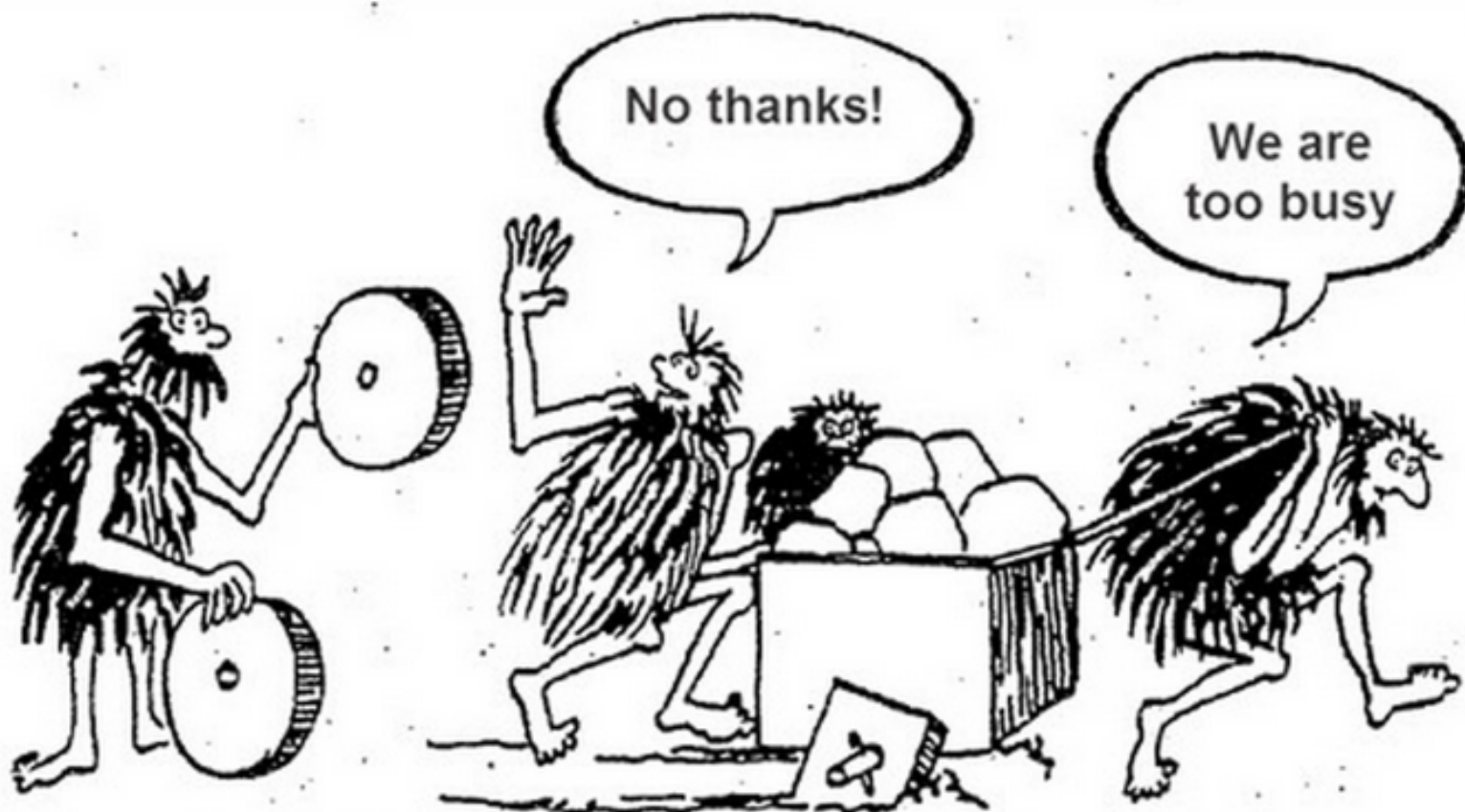
Comment:

Warnings:

Exceptions:

```
* throwable:
de.gsi.sequencer.demo.DemoException: something very bad happened
    at de.gsi.sequencer.demo.DemoSequence.lambda$3(DemoSequence.java:91)
    at de.gsi.sequencer.model.impl.Task.execUserCode(Task.java:54)
    at de.gsi.sequencer.model.impl.AbstractTaskImpl.call(AbstractTaskImpl.java:255)
    at de.gsi.sequencer.model.impl.AbstractTaskImpl.call(AbstractTaskImpl.java:1)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
```

space for sequence/task parameter



- ... check-out the demos first
- ... check-out the existing sequences/tasks
- KISS: – keep it simple and safe & document
 - think about what needs to be achieved first
 - keep the code readable (→ for your colleagues & yourself)
 - stick to Java (~ C++) coding styles
 - PMD and 'Find Bugs' Eclipse plugins are your friends!
 - stick to CO coding-style guide-lines!
- plan-ahead: Java Experts (<5%) vs. Non-Java Experts (the other 95% at GSI/FAIR)
 - 1. document sequence → code
 - 2. code → document sequence

- Common Task Interfaces:

```
// example of the most commonly used/useful setter methods
```

```
final TaskUserInterface dummyCode = (taskReference) -> {
```

```
    // my dummy user-level code code
```

```
};
```

```
final CallableTask myTask = addTask(new Task()).setTaskName("shortTaskName").setTaskDescription("???"  
    .setSequence(this).setSkippableState(true).setSkipTaskState(true).setTaskBreakPoint(true)
```

```
    .setTaskRepetitionNumber(5).addReference("task info reference, e.g. URL")
```

```
    .addUpdateListener((updateEvent) -> {
```

```
        // call me if you want to be notified about task execution state update
```

```
        // (usually only used within Sequencer service)
```

```
    }).setUserCode(dummyCode).setErrorRecoveryUserCode(dummyCode).setPreUserCode(dummyCode)
```

```
    .setPostUserCode(dummyCode);
```

```
myTask.isSkippable(); // getters mirror usually setters etc.
```

```
[..]
```