A framework for cut-over management

Guido Nageldinger

The purpose of this paper is to provide a governance structure for IT-related projects in order to assure a safeguarded and timely transition to a productive environment. This transitioning, which rarely exceeds a weekend, is colloquially called 'cut-over', 'rollout' or 'deployment'. The governance structure is defined in accordance with a set of project-specific deliverables for a cascade-type procedural project-management model, which is integrated within an Information Technology Infrastructure Library (ITIL)-orientated service organization. This integration is illustrated by the use of a semi-agile release model. Due to the release model selected, which is particularly characterized by its bundling of projects for a release-specific rollout (as it is referred to in the project documentation), a new definition and interpretation of deployment from a generic ITIL perspective is required. The facilitated release model requires a distinction between a *project-specific cut-over* and a *release-specific rollout*. This separation gives rise to two types of go-live scenarios: one for each participating project and one for each release. Additionally, an interplay between cut-over planning for a project and rollout planning for a release becomes apparent. Projects should already incorporate cut-over related deliverables in the initial planning phase. Even though consulting methodologies such as ASAP (Accelerated SAP), recommend scattered, project-specific deliverables useful for cut-over planning, this publication offers an integrated approach on how to prepare systematically for a project-specific cut-over with all required deliverables. The framework provided maps out ITIL's release and deployment process by means of IT projects; furthermore it allows IT projects to interface easily with the ITIL change-management process .

1 # A Framework for Cut-over Management

2 *Guido Nageldinger*[a]

3 [a]*Otto (GmbH & Co KG) · Depart. of Testing – and Release Management · Werner-Otto-Straße 1-7*

4 *22172 Hamburg · Germany · guido.nageldinger@ottogorup.com*

6 **Abstract**

7   The purpose of this paper is to provide a governance structure for IT-related projects in

8 order to assure a safeguarded and timely transition to a productive environment. This

9 transitioning, which rarely exceeds a weekend, is colloquially called 'cut-over', 'rollout' or

10 'deployment'.

11   The governance structure is defined in accordance with a set of project-specific

12 deliverables for a cascade-type procedural project-management model, which is integrated

13 within an Information Technology Infrastructure Library (ITIL)-orientated service organization.

14 This integration is illustrated by the use of a semi-agile release model. Due to the release model

15 selected, which is particularly characterized by its bundling of projects for a release-specific

16 rollout (as it is referred to in the project documentation), a new definition and interpretation of

17 deployment from a generic ITIL perspective is required.

18   The facilitated release model requires a distinction between a *project-specific cut-over*

19 and a *release-specific rollout*. This separation gives rise to two types of go-live scenarios: one for

20 each participating project and one for each release. Additionally, an interplay between cut-over

21 planning for a project and rollout planning for a release becomes apparent. Projects should

22 already incorporate cut-over related deliverables in the initial planning phase.

23   Even though consulting methodologies such as ASAP (Accelerated SAP), recommend

24 scattered, project-specific deliverables useful for cut-over planning, this publication offers an

25 integrated approach on how to prepare systematically for a project-specific cut-over with all

26 required deliverables. The framework provided provided maps out ITIL's release and deployment

27   process by means of IT projects; furthermore it allows IT projects to interface easily with the ITIL

28   change-management process.

29   **1. Introduction**

30   Most IT-related projects, in particular implementation and software development projects,

31   change a productive system landscape when taken live. On the one hand these projects face

32   the challenge of delivering a change (within an ITIL context) in a timely and cost-effective

33   manner. On the other hand, IT organizations need to assure the integrity of the system

34   landscape and are required to minimize potential interference to the ongoing business, as

35   Service Level Agreements (SLAs) need to be fulfilled.

36   The Central Computing and Telecommunications Agency (CCTA), a government agency in Great

37   Britain, already developed the Information Technology Infrastructure Library (ITIL) at the end of

38   the 1980's. The latest update of ITIL V3 was published in 2011 which is frequently referenced as

39   ITIL 2011. Is it still relevant? Yes, because it helps bringing order to IT chaos. Proehl et al. (2013)

40   analyzed 255 articles in the field of IT Service Management. They also confirmed the work by

41   Shahsavarani and Ji (2011). Both studies found a growing number of published papers dealing

42   with the development of concepts, constructs, models, methods and implementations for

43   theory development. Performance issues in IT Service Management, justifications, and IT

44   Infrastructure Library topics are among the most popular topics of research. Only few articles

45   have used or developed theories. Marone et al. (2014) found organizations adopting ITIL

46   implemented more operational level processes than the tactical/strategic level processes. The

47   study utilizes 623 survey responses from the UK, USA, DACH (German-speaking countries) and

48   Australia.

49   ITIL is rarely seen in isolation and current research focuses on the integration with IT

50   governance standards, such as ISO/IEC 38500 and other methodologies, such as COBIT,

51   PRINCE2 and the ISO2700x-family (Tanovic and Orucevic, 2011; Shivashankarappa et al., 2012;

52   Jäntti et al., 2014; Larsson, Rusu & Aasi, 2015).

53   ITIL's release and deployment management processes are demanding. Jokela and Jäntti (2012)

54   as well as Lahtela and Jäntti (2011) identified within their case studies the following common

55   challenges:

56   • no existing process for product portfolio release and deployment management

57   • lack of communication

58   • unclear release and deployment management and/or product portfolio management

59   process roles

60   • lack of resources and time for product portfolio integration, testing and reviewing

61   • existing models and practices are not suitable for agile software development

62   • uncertainty about of the contents of release packages

63   • high release distribution frequency

64   • different and tailored release packages

65   • the change management is not up to date and

66   • no existing service catalog

67   The project-specific cut-over, as defined in more detail below, requires detailed planning, many

68   meetings and several agreements with all stakeholders involved. Which questions need to be

69   addressed? Here are just some of them:

70   • How should the project outcomes be transferred to operation?

71   • Which scenario provides the best compromise between time, cost and risk?

72   • What happens if the cut-over fails?

73   • How can the previous system condition or version be reinstalled in case cut-over fails?

74   • How can we ensure that projects start early enough, with all the necessary preparation

75   work?

76   • How can cut-over activities be aligned within an IT service organization?

77   • How do we measure the success of the cut-over – and how can we maximise it?

78   Unfortunately, answers and activities associated with these cut-over related topics are

79   frequently addressed too late, probably because the cut-over is one of the final steps in

80   projects and can therefore be hidden for a long time. If a cut-over and its associated

81   deliverables are not integrated within the overall Project Plan then projects are at risk of being

82   delivered too late. For this reason, there is an urgent business need to govern projects in such a

83   manner that they do not forget to cover cut-over related items. So, which items should projects

84   cover?

85   This question and many others are addressed by a 'framework for deployment management' as

86   referred to below in this document. The purpose of this paper is to provide a methodology

87   which assures a robust and timely go-live for projects.

88   An itemised objective overview:

89   • to list the key deliverables projects need to provide in order to assure a timely and safe go-

90   live

91   • to illustrate how these deliverables can be integrated within the ITIL release and

92   deployment process

93   • to look into different deployment types, which this paper refers to as (a) project-specific

94   cut-over, (b) application-specific cut-over and (c) release-specific rollout along with the

95   timeframes and timepoints involved, and lastly

96   • to present a framework which enables an IT-service organization to transition a project to a

97   productive environment.

98   The scope of the deployment framework introduced is required

99   • to be applicable within an Information Technology Infrastructure Library (ITIL)-orientated

100   service organization, and

101   • to extend commonly facilitated project-management methodologies.

102   As with most management practices, there are always more ways to achieve similar results with

103   other approaches. However, in order to examine and establish good management practice,

104   these methodologies need to be published to facilitate comparison. The framework published

105   here is just one of many potential options. It illustrates how deployment activities can be

106   harmonised with commonly facilitated IT projects as well as quality-management practices, and

107   integrated in an ITIL environment.

108   Even though it is possible to take advantage of the deployment framework presented here

109   without ITIL, many companies have adopted ITIL recommendations and best practices and are

110   organized in a service-orientated manner. IT projects are therefore unlikely to exist in isolation

111   and are more likely to be embedded in a service-orientated manner. ITIL is known to be generic

112   and by itself is not complete. Its purpose is not to provide advice on how to implement IT

113   Service Management processes. Instead it requires integration with other disciplines such as

114   project management and quality management.

115   The framework presented here in the form of an illustrative case study arose during my

116   consultancy work and has been implemented for some applications within the Department of

117   Testing and Release Management of Otto (GmbH & Co KG) which is a member of the otto

118   group. It is not a hypothetical model. Projects which wish to cut-over in form of a release-

119   specific rollout are requested to comply with this framework. Within every release between 15

120   to 25 projects are participating. Six release-specific rollouts are conducted every year. Even

121   though consulting methodologies such as ASAP (Accelerated SAP) recommend scattered

122   project-specific deliverables useful for Cut-over Planning, this publication offers an integrated

123   approach on how to systematically prepare for a project-specific cut-over with all required

124   deliverables. The framework provided extends the ITIL release and deployment process by

125   means of IT projects and allows IT projects to interface easily with the ITIL change-management

126   process.

127   The activities of the otto group, with its headquarters located in Hamburg, Germany, are

128   grouped into three main business areas: (i) Multichannel Retail, (ii) Financial Services and (iii)

129   Service. This structure is consistently reflected in the group's activities along the retail value

130   chain, in Logistics for instance. In the year 2014/15, the Group generated consolidated revenue

131   of more than 12 billion euros and employed about 54,000 staff worldwide.

132   **2. Project-specific cut-over versus release-specific rollout**

133   ITIL is often facilitated as a checklist. With regard to the release and deployment process, it

134   consists of the following steps (Rance, 2011):

135   • plan and prepare a release

136   • build and test a release

137   • plan and prepare deployment

138   • conduct deployment

139   • conduct review, and

140   • provide early-life support.

141   Due to its generic nature and the need to interface with other disciplines, IT-service

142   organizations implement these steps quite differently. The variability of implementation

143   options may be caused by different interpretations of the term *'release'*, which is also related to

144   the term *'change'*. How does ITIL define these terms?

145   The term 'release' is defined as: *"One or more changes to an IT service that are built, tested and*

146   *deployed together. A single release may include changes to hardware, software,*

147   *documentation, processes and other components."* (AXELOS, 2011)

148   The term 'change' is defined as: *"The addition, modification or removal of anything that could*

149   *have an effect on IT services. The scope should include changes to all architectures, processes,*

150   *tools, metrics and documentation, as well as changes to IT services and other configuration*

151   *items."* (AXELOS, 2011)

152   In order to safeguard the related system landscape as well as associated services, conventions

153   need to be outlined on the practical arrangement and organization of release-related changes.

154   Here, the term '**release-model'** is facilitated for a set of rules and conventions used to bundle

155   and organize a release.

156   **Figure 1** illustrates a semi-agile release model, which is used within this publication as an

157   example to address some dependencies (Nageldinger, 2015). This release model is called 'semi-

158   agile' here because it consists of an agile section and a release phase, which follows a classical

159   sequential pattern (cascade model). During the agile period, phases between projects are non-

160   clocked. Project phases which fall within the agile period relate to the Project Planning, design

161   and realization section, and can be conducted in sprints. Once these projects participate in an

162   integration test, then their phases need to be clocked. Independently of which release model is

163   facilitated, the release phase will most likely consist of (i) an entry gate, (ii) technical and

164   business-related integration tests with their acceptance procedures, and (iii) the release-

165   specific rollout (see **Figure 2**).

166   Let us now look at the deployment types. In case the bundling of projects is foreseen by the

167   release model, such as in the one presented here, then we encounter deployment related to

168   the release as well as deployment related to participating projects. ITIL does not distinguish

169   between these. Here, the term *'deployment'* (AXELOS, 2011) is defined as *"an activity*

170   *responsible for movement of new or changed hardware, software, documentation, process etc.*

171   *to the live environment. Deployment is part of the release and deployment management*

172   *process."* It can probably be argued that such a distinction is unnecessary, since all bundled

173   projects are to be deployed in the same time slot. However, the project-specific deployment

174   and its associated preparation work are owned by the Project Manager and, in the case of

175   larger projects, by a designated Cut-over Manager. According to ITIL the Release and

176   Deployment Manager owns the release and its associated deployment.

177   ITIL's definition of '**deployment**' lacks conceptual clarity. This is because if we look only at the

178   movement of software to the live environment then we also need to distinguish between the

179   actual movement and its use when the software is 'switched on'. This is necessary if legal

180   changes come into effect at a specific point in time, for example, but the software is required to

181   be moved to the live environment beforehand. For the purposes of this paper the term

182   '**deployment**' from now on is solely used to refer to the movement of software to the live

183   environment, which is commonly conducted within a restricted timeframe. The term '**point-of-**

184   **use**' is facilitated to account for the point in time in which the software is 'switched on'.

185   In order to safeguard our system landscape and associated services we need to monitor both

186   the deployment (timeframe) and point(s)-of-use (point in time). In the context of a release, one

187   deployment and potentially more points-of-use are scheduled. This aspect is elaborated further

188   below in the discussion of the release-change.

189   Besides the participating projects, a release additionally includes service packs (a phrase used

190   for minor upgrades), bug fixes and smaller system-related features. These non-project elements

191   are usually governed by separate changes. All ingredients of a release should provide the option

192   of independent transitioning to the productive environment. If this option is not provided, then

193   projects or non-project elements cannot easily be excluded from the release if they fail the

194   integration test. Smaller projects are likely to be bundled in form of a release, and then

195   transitioned to a productive environment in form of a release-specific rollout. Major IT

196   implementation projects are preferably transitioned to a productive environment

197   independently, in order to reduce complexity.

198   The term '**project-specific cut-over**' defines the transition of a project to a productive

199   environment. It relates to the project as a whole and includes the software deployment and its

200   point of use. The actual timeframe associated with the project-specific cut-over is called the

201   '**cut-over window**'. The '**release-specific rollout**' defines the collective transition of bundled

202   projects to a productive environment. In the same way, the associated timeframe is called the

203   '**Rollout Window**' (Nageldinger 2014).

204   Frequently, projects communicate what is frequently called a 'go-live date' on their overall

205   project chart. In view of the reasons given above, this could mean (i) end of the software

206   deployment, (ii) end of cut-over window, which relates to the transitioning of the whole

207   project, or (iii) point-of-use. It is therefore recommended to question what exactly this date

208   refers to.

209   Beside the project-specific cut-over and the release-specific rollout we are most likely to face a

210   third transitioning aspect, here called the '**application-specific cut-over**': this relates to a

211   specific system or application and mainly covers approaches related to maintenance or

212   upgrades. It differs from the project-specific cut-over in that it consists of canonical (i.e.,

213   reoccurring with every rollout) activities and tasks. The deployment of service packs can be

214   organized canonically, for example.

215   Within an SAP context the term 'rollout' is frequently associated with a so-called 'template

216   approach'. In this, the core configuration is embedded within a reusable template and the

217   country-specific characteristics are added separately in the form of a local configuration (SAP,

218   2009). SAP's usage of the term 'rollout' is quite similar to the definition provided here and its

219   associated release context. This is because the release-specific rollout also consists of

220   reoccurring activities, here referred to as "application-specific cut-over", which is similar to

221   SAP's template approach. The project-specific cut-over activities, which are non-reoccurring,

222   may be considered in the same way as SAP's country-specific characteristics.

223   The **Cut-over Schedule** is a list of tasks and activities required in order to conduct the cut-over.

224   It differs from the ordinary overall project schedule as it only focuses on tasks and activities

225   within the cut-over window, of short duration and lasting only for a couple of hours or a

226   weekend. The Cut-over Schedule is created by duration-based scheduling techniques.

227   Additionally, it contains a list of prerequisites, such as the delivery of training courses and the

228   installation of printers etc. These prerequisites need to be completed before the actual cut-over

229   can be conducted and are part of the overall project schedule. The Cut-over Schedule does not

230   relate to the post go-live phase. For this objective projects need to have a separate plan, which

231   for example addresses aspects of support, data-conservation and accounting closure

232   procedures.

233   Besides the Cut-over Schedule the Cut-over Plan contains a variety of other deliverables, in the

234   same way as a project-management plan, which are further elaborated below. Key elements of

235   a Cut-over Schedule are for example termination points, frequently called '**points-of-return**'

236   (POR). Theses PORs aim to trigger activities used (i) to restore the initial condition or (ii)

237   incrementally fall back to a previous POR. The last POR is referred to as the '**point-of-no-return**'

238   (PONR). The PONR does not always fall within the cut-over window. The Fallback Concept is the

239 name of a document used to describe activities which are potentially necessary at these PORs,

240 in case the system needs to be rolled back to a previous state or even its initial state. If the cut-

241 over activities pass the PONR then we enter the disaster scenario. For such events, a disaster-

242 recovery concept defining appropriate measures should be in place. The disaster-recovery

243 concept is mentioned here because it is unfortunately frequently neglected. It is regularly

244 associated with earthquakes, hurricanes or terrorist attacks. However, unforeseen events

245 during a cut-over after the PONR can also trigger the disaster scenario (BSI, 2008). The cut-over

246 window is illustrated in **Figure 3**.

247 The sections above clarify why certain release-models, which foresee the bundling of projects

248 for collective rollout, require at least two perspectives – a project-specific and a release-specific

249 one. An important outcome for the preparation of a project-specific cut-over and a release-

250 specific rollout is the go-live scenario. A go-live scenario defines the overall approach on how a

251 cut-over or rollout is to be conducted. The go-live scenarios differ in most cases, which is

252 illustrated by an example in **Figure 4**. The go-live scenario of project A utilizes the Rollout

253 Window of release 1 and 2, whereas projects B and C merely take advantage of one Rollout

254 Window. The release-specific rollout also facilitates go-live scenarios. Here, the rollout of

255 release 1 uses a sequential rollout strategy, while the rollout of release 2 uses a parallel rollout

256 strategy.

257 **3. Recommended deliverables and outcomes for the preparation of a project-specific cut-**

258 **over**

259   This chapter covers recommended deliverables and outcomes for projects in order to prepare

260   for a project specific cut-over. Projects are assumed to be arranged in a cascading manner with

261   classic phases, in order to keep this publication as simple as possible. However, as stated above,

262   activities may be undertaken in an agile manner before entering the release phase.

263   The release model presented in **Figure 1** illustrates the following generic phases, which are

264   quite similar in the case of IT-related projects:

265   • Project-planning phase

266   • Design phase

267   • Realization phase

268   • Test phase

269   • Cut-over phase

270   • Support phase.

271   Quality Management is facilitated in order to request these cut-over relevant project

272   deliverables, which are then reviewed by Rollout Management. **Table 1** illustrates potential

273   deliverables (Nageldinger 2015). Some of them are mandatory, others are optional. These

274   deliverables are further elaborated here below and presented according to their project

275   phases.

276   *3.1.   Project-planning phase*

277   The preparation of a project-specific cut-over already needs to be considered during the

278   planning phase of a project. Key deliverables are (i) a decision related to the establishment of a

279   Cut-over Manager position, (ii) an agreement on cut-over specific deliverables, which are

280   included (iii) within a plan for Deployment Management.

281   *3.1.1. Plan for Deployment Management*

282   In order to deliver a Cut-over Plan on time it is important that the cut-over related deliverables

283   and outcomes are already taken into consideration during the planning phase of a project.

284   Larger projects will probably require a separate schedule for all cut-over related planning tasks

285   and activities, whereas smaller projects include them in their overall project schedule. Here,

286   this key deliverable is called deployment management plan. Please note that these tasks relate

287   solely to the necessary preparation and planning work and not to the Cut-over Schedule itself.

288   *3.1.2. Establish Cut-over Manager (COM) position*

289   During the planning phase it needs to be decided whether a Cut-over Manager (COM) position

290   is to be implemented. The COM role may be compared with the role of a Project Manager; it is

291   solely an administrative role which covers IT-related topics as well as business related ones. The

292   only difference is that the COM focuses on the preparation work associated with activities

293   conducted within the cut-over window. A COM is responsible for the preparation, planning and

294   execution of the project-specific cut-over. He or she needs to be positioned beside the Project

295   Manager and should be part of the Steering Committee. This is because conflicts of interest are

296   likely to occur between the COM and the Project Manager. Additionally, a COM needs to drive

297   decisions and potential change-requests related to the cut-over; many of these bear associated

298   risks. A common role for a COM would include the following responsibilities:

299   • drive a decision on the final go-live strategy

300 • implementation and guidance of a Cut-over Team

301 • creation of a Cut-over Plan

302 • creation of a list with initial conditions required to be met prior to the cut-over

303 • Initiating a list of activities after cut-over

304 • Planning, execution and organization of the cut-over-test and go-live-simulation.

305 Smaller projects do not usually need a COM as the Project Manager fulfils this role.

306

307 *3.1.3. Initial dialogue with Rollout Management*

308 During the project-planning phase an initial dialogue with Rollout Management is mandatory

309 because related deliverables need to be scaled to the project size. This scaling is quite difficult

310 to automate since it depends on experience and the potential interfaces involved. A simple

311 formula, for example based on the project budget, is insufficient.

312 *3.2. Design phase*

313 Prior to completion of the design phase, the Go-live Strategy and Fallback Strategy should be

314 finalized. Depending on the project size, these key deliverables can be combined and should

315 provide a top-level description of options on how to transition the project to a productive

316 environment, as well as how the initial state can be restored.

317 *3.2.1. Go-live strategy*

318   The Go-live Strategy is of an investigative nature. It should provide several potential go-live

319   options with their associated impacts, rather than an ultimate solution. It should foster

320   discussions related to the topic. The idea is to provide a general overview and associated

321   opinions. This document is later used to balance stakeholder interests and to carve out the final

322   go-live method.

323   A Go-live Strategy starts with the go-live scope. How does the go-live scope differ from the

324   project scope? Practically speaking, the *go-live scope* is the elaborated version of the initial

325   project scope and it is therefore more concrete. The *project scope*, for example, can state the

326   introduction of a system to meet the purpose XYZ. A go-live scope needs to state precisely

327   which system is going to be implemented, lists the technology involved and states the required

328   organizational changes. A project scope, for example, can state the improvement of a process.

329   The go-live scope needs to state how a particular process is to be improved and which changes

330   are required to be conducted. It is recommended to define the go-live scope in writing during a

331   workshop attended by all project key players. This exercise is valuable for the overall project, as

332   well as in identifying scope changes already occurring. A properly defined go-live scope is a

333   prerequisite for evaluating all other cut-over relevant questions.

334   The Go-live Strategy covers the following aspects:

335   •   a description on how the project can be transitioned to the live environment. The

336   description can be of generic nature and should relate to the major approaches, such as

337   big-bang, phased go-live, or iterative methods.

338   •   It is highly recommended to provide several go-live options which identify and provide

339   for the various associated risks.

340   •   Many go-live scenarios potentially impact the business and technical infrastructure. The

341   impact and its potential consequences should be discussed with the stakeholders. Their

342   opinion should be included within the Go-live Strategy and referenced as well.

343   •   Alternative go-live approaches, which are probably not investigated further, should be

344   defined and recorded. Additionally, it should be recorded why particular options are not

345   followed up.

346   •   One section should focus on potential go-live dates. Again, rather than providing one

347   ideal date, several potential dates should be evaluated. What are the benefits and

348   catches of these dates? Which dates could be used as an alternative, in case of project

349   delays? Who recommends these dates? What is the reasoning behind them?

350   •   A cut-over window usually last for just a few hours. However, some projects require a

351   weekend. Which constraints and risks are associated with the cut-over timeframe?

352   What are the stakeholders' wishes? Why do they request a shorter or longer

353   timeframe?

354   •   Many projects alter existing *business processes*. These changes should be addressed in

355   form of an AS-IS and TO-BE description. Which processes are newly introduced, altered

356   or deactivated? Departments which are impacted should be identified as well. Who are

357   the key contact persons?

358   •   Some projects alter *the organization*. If this is the case, what does the new organization

359   look like? A section of the Project Plan should describe and illustrate the old and new

360   organization. It should highlight the targeted changes, as well as how the targeted

361   changes are to be achieved. Are the change constraints related to the timeframe?

362   Projects with significant reorganization focus should employ an organizational change

363   management task force which delivers this information.

364   • Since people tend to change their opinion or might need to be contacted later for this,

365   further discussions on all related references should be logged.

366   • How many people are going to be impacted during and after a potential go-live? The

367   impact can be of a technical, organizational and / or business nature.

368   It is an open secret that many companies do not document their business processes. Major

369   implementation projects therefore require this task to be conducted. The Go-live Strategy

370   needs to focus on the targeted change by comparing the AS-IS with the TO-BE state. It outlines

371   transition options in order to cut-over a project to a productive environment. These transition

372   options also include the technical aspects and system landscape. However, the technical

373   documentation of the AS-IS and TO-BE is usually conducted in a separate document, here called

374   a 'System Landscape Design' (SLDe) document.

375   *3.2.2. System Landscape Design (SLDe)*

376   The creation of an SLDe is commonly the task of a core technology or IT architecture

377   department. For further Cut-over Planning it is obviously essential that this document exists,

378   and that the current as well as the future system landscape is not only presented in a graphical

379   manner but also properly described. Even though some companies have a Configuration

380   Management Database (CMDB) in place, the information available is frequently not

381   newsworthy or lacks information required by the project. The creation of an SLDe is therefore a

382   task which needs to be foreseen in most IT implementation projects. From the cut-over

383   perspective additional questions need to answered, for example:

384   • who are the applications contact persons (such as Business Owner, Administrator, key

385   users etc.)?

386   • Which business process are linked to the AS-IS and TO-BE system landscape?

387   • Which departments currently work with the applications involved?

388   • A list of associated applications

389   • A list of interfaces, their business objective and the technical protocol used.

390

391   *3.2.3. Extension to SLDe*

392   Certain go-live approaches require an extension of the SLDe, which is treated here as a separate

393   deliverable in order that it is not forgotten. Extensions are, for example, particular interfaces for

394   data-migration purposes. System landscapes related to the proposed test methodology as well

395   as the go-live simulation and the cut-over test, which are explained further below, are also

396   important extensions of the SLDe.

397   *3.2.4. System profile description*

398   If a project is initiated in an environment where a CMDB is missing or not sufficiently

399   maintained, then system-profile descriptions need to be created; this is commonly conducted

400   in the form of a survey. Such a survey should be conducted together with the department

401   which already owns the CMDB. It is therefore advisable during the project-planning phase to

402   conduct a brief review on how well the system landscape is documented and to foresee the

403   system-profile descriptions as an evaluation task within the project schedule, since this task can

404   become quite time-consuming.

405   *3.2.5. Fallback Strategy*

406   The Fallback Strategy should be defined in writing within the same timeframe as the Go-live

407   Strategy. It provides a top-level draft of the potential activities required in order to reinstall the

408   initial state prior to the start of cut-over. The detailed version of this document, here called

409   'Fallback Concept', will be created later and finalized during the test phase. The Fallback

410   Strategy addresses the following questions:

411   • how can the system landscape be rolled back to its state prior to the start of cut-over?

412   Are further scenarios possible?

413   • How long would such a fallback procedure take?

414   • Which criteria or conditions should trigger a fallback?

415   • Which risks are associated with each fallback option provided? What do their mitigation

416   strategies look like?

417   • In case of a fallback, how great is the potential business impact? Can the potential

418   impact be quantified? How great is the impact in case a foreseen fallback scenario fails?

419   How likely is it to occur? How great is the impact in case the system landscape involved

420   is unavailable for one day? How many days can a business survive without IT?

421   • Which model should be used in order to calculate the business impact?

422 • What is the maximum acceptable downtime? How is it calculated? Are Service Level

423 Agreements (SLAs) jeopardized? How great are the potential contractual penalties?

424 • Where can potential PORs be positioned? Where is the PONR? Do all relevant systems

425 have a disaster recovery concept in place, in case the PONR has been reached and things

426 go wrong? Is it possible to assign a potential business impact to each POR identified?

427 What are the termination criteria associated with each POR?

428 The questions provided are obviously not complete, but are intended to illustrate the potential

429 complexity involved.

430 *3.2.6. Communication-related deliverables*

431 Cut-over Management is communication management, and related deliverables such as a

432 Communication Plan, a stakeholder analysis and a kick-off, to mention just a few, do not differ

433 from ordinary project-management methodologies and are therefore not further elaborated

434 here. However, these methodologies need to be applied within the cut-over context. A

435 warehouseman, for example, is unlikely to be part of an ordinary project stakeholder analysis.

436 However, within a cut-over context, a warehouseman can be an important key player. By the

437 end of the design phase the Cut-over Team should be defined. The Cut-over Team consists of all

438 people required to contribute to the cut-over deliverables explained within this section. The

439 Cut-over Team also should be completely defined by the end of the design phase.

440 Communication-related deliverables are recommended not to be evaluated during a potential

441 assessment. This is because both a Go-live Strategy and a Fallback Strategy require these

442   communication items to be in place in order to prove that all-important stakeholders' views

443   have been considered.

444   The framework for Cut-over Planning can also been seen as a communication-related

445   deliverable. It is a document which describes how planning for the cut-over is to be conducted,

446   and which cut-over related deliverables are to be produced. Such a document is obviously only

447   advisable for larger cut-over tasks and is therefore not mandatory.

448   *3.3.   Realization phase*

449   *3.3.1.   Decision related to the go-live and Fallback Strategy*

450   The decision related to the go-live and Fallback Strategy is a key outcome of the realization

451   phase. It needs to be conducted by the Steering Committee since it relates to the triple

452   constraints, such as scope, timeframe, budget and risk. The decision is required to be prepared

453   in form of a decision-model presentation and commonly extensively discussed, prior to a

454   Steering Committee meeting. It is then up to the members of the Steering Committee to

455   consult or involve further senior management if the go-live risks associated require this. This

456   decision may present several political challenges and if it is conducted too late then the timely

457   delivery of the project is at risk.

458   *3.3.2.   One-pager*

459   The One-Pager summarizes the go-live and Fallback Strategy decided. It is later integrated

460   within the release-specific Rollout Handbook. The objective of the one-pager is to inform all

461   other projects and participating parties about the rollout and to facilitate the preparation of the

462   release-specific rollout.

3.3.3. Draft Cut-over Plan

The Cut-over Plan and the Cut-over Schedule are not the same and are explained further below, within the context of the Rollout Handbook. The Cut-over Schedule is an important part of the Cut-over Plan. As a minimum requirement a draft Cut-over Schedule should be available at the end of the realization phase, sufficient to support the planning activities for the release-specific rollout. Since the term 'draft' can be interpreted in several ways, it is suggested to define the outcome in advance in order to avoid disappointment at the end of the realization phase. This draft schedule should contain all activity blocks, with their subordinated activities as well as their durations, dependencies within these activity blocks, with an attached working breakdown structure (WBS) and a Glossary.

3.3.4. Risk Register

The cut-over related Risk Register does not differ much from an ordinary project Risk Register. The administrative work of this Risk Register should be kept to a minimum, since the risks solely relate to the cut-over. However, it is still necessary that such a record is properly administrated. Since key elements of such a Risk Register are quite frequently forgotten, such as the Risk Owner or Risk Indicator, a practical set of parameters are suggested here, such as:

- **Risk ID:** the ID number assigned to the risk
- **Risk Category:** a sorting criterion used to categorise risks. Commonly used categories are, for example, applications or phases during a rollout.

482 · **Risk Description:** this can follow a simple scheme by addressing three questions: (i)

483 · What is the actual risk? (ii) How is the risk caused? (iii) What is the impact if the risk

484 · does occur?

485 · **Alerter:** can be anyone who identifies and reports a risk.

486 · **Alerting Date:** the date on which a risk has been recorded.

487 · **Magnitude:** calculated on the basis of the probability and potential impact of a risk

488 · occurrence.

489 · **Risk Indicator:** criteria used to recognize that a risk has manifested itself; for instance,

490 · error messages after a database update etc.

491 · **Mitigation Measure:** a counter-action that can be taken if a risk manifests itself.

492 · **Risk Owner:** a person with the assigned authority and competency to mitigate a risk.

493 · The Risk Owner identifies mitigation strategies, defines Risk Indicators and nominates a

494 · Technical Contact for the respective rollout. This role also includes refining the Risk

495 · Description, since the risk as such is frequently confused with the causes of a risk.

496 · **Technical Contact:** a technical expert or a team which shares the required technical

497 · expertise and is in any case present during the rollout, usually fulfils this role. This

498 · person is responsible during the rollout (i) to verify whether a risk has manifested itself,

499 · and (ii) to coordinate Mitigation Measures defined within the Risk Register.

500 · **Processing Status:** describes how far the processing of the risk has progressed.

501 · o **New:** a new risk has been recorded, described and assigned to an initial Risk

502 · Owner;

503   o **Under Way:** the Risk owner has accepted his role for the particular risk raised,

504   and communicates a delivery date on which the Mitigation Measures and Risk

505   Indicator are defined;

506   o **Completed:** as soon as the Mitigation Measures and Risk Indicator are described

507   and a Technical Contact for the particular risk has been nominated, Risk

508   Processing has been completed.

509   • **Completion Date:** the date on which processing status is set to 'Completed'.

510   *3.3.5. Dependency Matrix*

511   A collective participation by projects in a release-specific rollout causes a need to manage

512   dependencies. These dependencies can occur between Project Teams as well as jointly used

513   personnel resources, here called 'Core Teams'. The Dependency Matrix is a tool used to

514   evaluate the potential impact and interrelation between Project Teams and Core Teams.

515   Project Leads are requested to fill out a template and to identify which key personnel are

516   required to be present during the cut-over.

517   The Dependency Matrix is hard to define as a tool because the terms (i) 'Core Team' as well as

518   (ii) 'dependency' are themselves difficult to define. Here, a Core Team is seen as a

519   heterogeneous group of key personnel required to be present during a cut-over/rollout. It

520   consists, for example, of administrative personnel related to key applications and databases as

521   well as Service Owners (within an ITIL context). A dependency between a Project Team and a

522   Core Team is present if communication between the Project Team and the Core Team or its

523   related technology is necessary during cut-over/rollout. Since Project Teams usually identify

524   similar Core Teams, a template can be created. This is illustrated in **Figure 5**. The bold points

525   visualize dependencies between Project Teams and Core Teams participating in a release.

526   Even though the term 'Core Team' is treated less strictly, these dependencies can be facilitated

527   to quantify the complexity of a release. Complex projects usually require more key personnel

528   than less complex projects. A complex release usually consists of several projects with many

529   dependencies. Therefore, one possible measurement of the complexity of a release is the total

530   amount of dependencies. This idea is further elaborated below. Additionally, such a

531   Dependency Matrix is quite a sensible instrument to have in order to identify projects which

532   have probably influenced a particular business process negatively after rollout.

533   *3.3.6.   Dialogue with Rollout Management*

534   Projects are requested to pass a gate prior entering the release phase, which consists of a

535   variety of project deliverables. Most of the cut-over related deliverables have already

536   presented here, such as (i) the decision related to the Go-live Strategy, (ii) the Cut-over Plan

537   (draft), (iii) a Risk Register, (iv) Dependency Matrix and (v) an obligatory dialogue with Rollout

538   Management. The cut-over related deliverables are discussed during the dialogue along with a

539   couple of questions which the Rollout Manager is to address.

540

541   *3.4. Test phase*

542   During the test phase the Cut-over Plan and Fallback Concept need to be finalized. The disaster

543   recovery concept relates to the period after the Point-Of-No-Return (PONR) and is required to

544   be produced for new systems or potentially extended for existing systems. It has already been

545   mentioned that an insufficiently documented system landscape can severely affect a project's

546   progress. It should therefore already been evaluated during the Project Planning phase whether

547   sufficiently documented disaster recovery concepts are in place for existing systems.

548   After completing the integration tests, a cut-over test and a go-live simulation should be

549   conducted. The cut-over test is usually the last test prior to the rollout with the objective of

550   validating the Cut-over Schedule. Some steps are simply conducted in an exemplary fashion,

551   such as data migration. This is in particular valid if a huge amount of data is required to be

552   migrated. A cut-over test would also cover a couple of fallback scenarios and (if necessary) a

553   disaster recovery case as well. A second form of validation is what this paper refers to as the

554   'go-live simulation', which has more similarities to a rehearsal than a test. The go-live

555   simulation focuses on the communication requirements during a cut-over and necessary

556   delivery points. It should be conducted with the whole Cut-over Team (personnel involved in

557   the cut-over) in place under circumstances which are close to the real cut-over situation. For

558   this reason the plan for the deployment of personnel should be in place as well. Both the cut-

559   over test as well as the go-live simulation can be conducted solely for a particular project

560   and/or together with other projects as part of a release. The scope of both validation forms

561   needs to be defined as part of the decision related to the Go-live Strategy during the realization

562   phase. The effort required to conduct these tests is quite significant, since IT landscapes close

563   to the live environment are required.

564   *3.5. Cut-over and support phase*

565   During the cut-over phase the operative cut-over is to be conducted either with other projects

566   in form of a release, or by one project alone. In the latter case, the project does not participate

567   in a release and conducts a project-specific cut-over independently. Such an independent cut-

568   over is most likely conducted in large-scale projects.

569   ITIL recommends a support phase as part of the release and deployment process (Rance, 2011).

570   However, support can only be provided by the personnel who have supporting knowledge, that

571   is, knowledge which can be provided exclusively by the Project Teams and not by the personnel

572   owning the release and deployment process. It is therefore argued that after the end of the

573   Rollout Window, the subsequent support phase is required to be covered by the Project Teams.

574   It is advisable to incorporate a specific period for the observation of incidents after cut-over or

575   rollout prior to the Lessons-learned Workshop. This observational period lasts from 1 to 4

576   weeks and obviously depends on the complexity of the project or release. This aspect is further

577   developed below, together with recommendations on how to run a Lessons-learned Workshop.

578   **4.   Recommended Deliverables and Outcomes for the Preparation of a Release-Specific**

579   **Rollout**

580   This chapter highlights a couple of important deliverables and outcomes of the release-specific

581   Rollout Management. Large-scale projects which are likely to cut-over independently may

582   consider the outcomes described here as well, even though these are assigned to the release

583   phase.

584   *4.1. The Rollout Handbook*

585   The planning document and key deliverable for the release-specific rollout is the Rollout

586   Handbook. This may be compared with the Project Plan defined in the PMBOK®. However, it

587   relates solely to the release-specific rollout. The PMBOK® (PMI, 2011) defines a Project Plan as

588   a "… formal, approved document used to guide both project execution and project control. The

589   primary uses of the Project Plan are to document planning assumptions and decisions, facilitate

590   communication among stakeholders, and document approved scope, cost and schedule

591   baselines. A Project Plan may be summarized or detailed."

592   Even though the Rollout Handbook is structured to the same degree as a Project Plan, it is only

593   finalised shortly prior to the actual rollout. A classic Project Plan, however, needs to be

594   completed at the beginning of the project, during the planning phase.

595   The Rollout Handbook consists of the following items:

596   **Executive Summary:** describes the major rollout activities, participating projects and

597   applications, highlights major risks and provides an overview chart of the Rollout Windows.

598   **Version History:** the Rollout Handbook is to be extended and adjusted throughout the whole

599   release phase. It is therefore important to publish temporary versions of this book in regular

600   iterations, so that it can be reviewed. A Version History is therefore crucial in order to keep

601   track of these changes.

602   **Communication Plan:** lists all participating persons with their contact details. Communication

603   related to the rollout is part of the Rollout Schedule and describes who needs to be contacted

604   when and with what information. The Communication Plan also includes the plan for the

605   deployment of personnel in all participating Project Teams and Core Teams.

606 **Risk Register:** aggregates the Risk Logs of all participating projects into a simple Excel format.

607 The format of the Risk Log is equal to the one described above.

608 **Description of Rollout Window:** is usually accompanied by a visual presentation and

609 summarizes major activities such a data migration or deployments on specific systems. An

610 important item of this description is a table including all participating projects, their associated

611 deployment times and points-of-use.

612 **One-Pager of all participating projects:** defines the scope of the rollout.

613 **Schedule Management:** a Rollout Schedule is quite similar to an ordinary project schedule;

614 however, some distinct differences exist. IT-implementation projects usually have a timeframe

615 of several months and are referred to ordinary projects in this paper. The release-specific

616 rollout only lasts for a couple of hours. Ordinary project schedules are usually managed with

617 duration-based or effort-based methods. A rollout can only be scheduled by duration-based

618 approaches, since the duration of most activities is fixed. Here, the duration of tasks is usually

619 scheduled on an hourly/minute basis.

620 Rollout Schedules need to have activity IDs and not only row numbers. What is the difference?

621 Scheduling is commonly conducted with planning tools which facilitate row numbers. The

622 planning tools use these line numbers for various calculations. However, they are also likely to

623 be changed quite frequently if the plan is changed. In order to avoid communication issues,

624 particularly during the operative rollout, alphanumerical IDs should additionally be used.

625 It is recommended to assign activity types to activities. Why? Firstly, Rollout Schedules are very

626 likely to be reused. This reusability is obviously much easier if unique activities, those that are

627   just used once for a particular rollout, are sufficiently labelled. Additionally, a Rollout Plan is

628   much easier to read if the operator knows already in advance, for example from the use of

629   particular colours, what types of activities are expected, such as communication activities or

630   checkpoints.

631   *4.2. Communication during Release Planning*

632   Preparation work related to the release-specific rollout mainly covers coordination meetings

633   with Project Teams and associated Core Teams participating in the release. Additionally, a large

634   number of staff is required to be informed about the status related to the rollout preparation

635   within the company. A rollout newsletter is a fit-for-purpose communication tool which can be

636   easily customized to cover the coordination aspect and meet the corporate information need.

637   The release model illustrated here is used in such a manner that 6 rollouts are conducted on an

638   annual basis, which results in an approximately 8-week preparation cycle for every rollout. A

639   typical schedule related to the planning activities and publication dates of such a release cycle is

640   illustrated in **Figure 6**. The newsletter is published about every other week. It comprises topics

641   related to the rollout preparation work, as well as subjects related to general IT questions such

642   as new documents and procedures published by the security personnel, changes conducted by

643   the Change Management etc. The newsletter is 2 to 6 pages long and announces deadlines and

644   important dates, such as the date for the approval of the Rollout Handbook, which is done

645   during a meeting with all key personnel. Documents related to the newsletter, such as the

646   Rollout Handbook, are stored on MS SharePoint®. This provides the desirable function of

647   enabling distribution of the newsletter to a wider community and limiting access to more

648   confidential documentation.

649   The 4th edition is published a couple of days prior to rollout. By this stage the Rollout Handbook

650   is in a final stage and everybody involved has a last chance to incorporate necessary changes.

651   The newsletter publishes the major steps involved during the rollout, key contact phone

652   numbers and an overview illustration. All interested parties thus know the overall picture of the

653   rollout, whereas access to the Rollout Handbook is restricted. The 4th edition publishes all major

654   planning steps and dates for the preparation work of the following rollout as well.

655   The 1st edition focuses on the incidents after the rollout. Incidents are observed up to 2 weeks

656   after the rollout. The observation period closes with a Lessons-learned Workshop, the

657   submission of the final report and the closure of the Rollout Change (see further below for the

658   treatment of the Rollout Change).

659   The 2nd and 3rd editions provide temporary planning results and draft versions of the Rollout

660   Handbook. Planning is conducted in three phases: during the 1st phase, interviews are

661   conducted between Rollout Management and Project Teams, see **Table 1** for the deliverable

662   'Dialogue with Rollout Management'. During the 2nd phase, Rollout Management interviews the

663   Core Teams, and throughout the 3rd phase fine tuning of the Rollout Handbook is carried out.

664   *4.3. Rollout Change*

665   There is no universal measure on how to organize changes. Obviously, IT-service organizations

666   need to record changes related to their system landscape; however, the way these are

667   recorded, the level of detail and types of changes used differ significantly between

668   organizations. What is a Rollout Change? The Rollout Change relates to the changes caused by

669   the release-specific rollout. However, it only covers the release-specific Rollout Window. , It has

670   already been highlighted above that a project can cause damage in potentially two ways: firstly

671   during the software deployment and secondly at the point-of-use. If the point-of-use falls

672   outside the Rollout Window then it is not covered by the Rollout Change. In this case an

673   additional change is required to be recorded. For this reason the Rollout Handbook contains a

674   register which lists the deployment window and point-of-use for every project. The Rollout

675   Handbook is appended to the Rollout Change as well.

676   The Rollout Change is created at the beginning of the release phase and then iteratively

677   extended by updating the attached Rollout Handbook. Once this Handbook has been approved

678   or is present in its final format, the change is requested to be approved by the Change Advisory

679   Board (CAB). The Change is closed after the completion of the Lessons-learned Workshop,

680   described in the next section.

681   *4.4. Lessons-learned Workshop*

682   Above it has already been explained that a Lessons-learned Workshop should be conducted

683   after completion of the rollout. Lessons-learned Workshops provide many generic benefits,

684   such as:

685   • the reuse of knowledge throughout an organization

686   • the reduction of project cost

687   • the improvement of methods and practices, and

688   • the enhancement of project delivery and quality management

689   just to mention a few of them. However, the Lessons-learned Workshop mentioned here

690   relates solely to the rollout conducted and is not of generic nature. Let us step back a little to

691   add some higher-level perspective.

692   Any IT organization has a need to improve its processes, ensure agreed service levels and justify

693   its projects or initiatives. For this reason, Continuous Service Improvement (CSI) represents one

694   of ITIL's five process groups. The CSI Register, which lists all improvement options and

695   initiatives, is one of an IT organization's key deliverables (Cabinet Office, 2011). Key

696   Performance Indicators (KPIs) are used to measure the process. How can we measure the

697   success of a rollout? What do we mean by success? ITIL publications already suggest a variety of

698   KPIs, such as the number of incidents caused after the release-specific rollout. These simple

699   KPIs can, when implemented in practice, be quite tricky. Even though this particular KPI is

700   suggested for the service validation and test process within ITIL's service transition process

701   group (Rance, 2011), incidents are always a good indication of disturbances and are here also

702   used for the evaluation of the rollout's success. **Figure 7** illustrates the total number of

703   incidents recorded within one IT service organization. The dotted lines indicate the week during

704   which a rollout occurred. The sharp decline of incidents during CW 30 is because a service-

705   request process was introduced. The dips between 200 to 500 incidents/week are caused by

706   the Christmas season, where many people are on holiday. Otherwise, it is quite difficult to

707   acknowledge any impact of the rollout on the total number of incidents recorded. The picture

708   changes once we take a closer look at a particular system. **Figure 8** illustrates second-level

709   incidents for a main order-handling system, which participated in all rollouts. These incidents

710   account for about 5% of the total number of incidents recorded. After some rollouts a rise of

711   incidents can be observed. How should we measure this?

712   **Figure 9** shows the variability of incidents four weeks before and after rollout. The connecting

713   lines are to aid visualisation of the form of the curves and are not intended to suggest data

714   points at all. The data provided are normalized. Here, the mean value of incidents during a

715   period of four weeks before rollout is set to 100%, with a standard deviation of +/- 18%. The

716   figure suggests a significant increase of incidents 1-2 weeks after rollout. This increase is

717   summarized in **Figure 10**, where the average increase in incidents two weeks after rollout is

718   plotted. Most rollouts show a significant increase. The negative value occurred during the

719   summer holiday period time, when less people recorded and reported incidents.

720   The values provided have been incorporated for illustrative purposes. It is hardly surprising that

721   if the number of projects participating in a rollout increases, then the number of incidents

722   increases as well. **Figure 11** shows a correlation coefficient of 0.4 between the average increase

723   of incidents two weeks after rollout and the number of participating projects per release. If we

724   consider the complexity of a rollout as introduced above, see also **Figure 5**, which is the total

725   number of interactions between Project Teams and Core Teams participating in a rollout, then

726   we obtain a similar correlation coefficient of 0.36. The high level of incidents provides a good

727   primary source of information on how well the rollout and release performed. However, these

728   data should only be used indicatively and need to be interpreted carefully, as:

729   •   processes or the ways processes are managed change over time, such as the above-

730   mentioned introduction of a service-request process, which reduced the number of

731   incidents

732   •   some projects separate their cut-over and point-of-use. Incidents can, for example, pop up

733   two or three weeks after rollout once the software goes live

734   •   the number of incidents recorded is impacted by seasonal variability, such as holiday

735   periods and Christmas

736   •   changes are conducted after and prior to rollout, and can significantly influence the

737   statistics.

738   In order to learn what actually caused these incidents they need to be examined in a more

739   detailed manner, which is the purpose of the Lessons-learned Workshop recommended here.

740   The Lessons-learned Workshop should be attended by all key personnel involved, such as the

741   Project Manager/Cut-over Manager of the participating projects, Test Manager, and staff

742   involved in the resolution of 2nd and 3rd level incidents – just to mention some of them. In order

743   to run such a workshop it is essential to identify the incidents/service requests most frequently

744   caused by the rollout and to classify them. These incident categories are used solely for rollout-

745   evaluation purposes and additionally to the ordinary classification scheme, which is part of the

746   incident-management process.

747   **Table 3** provides an example of typical incident categories used after rollout evaluation.

748   Between 60-80% of the incidents evaluated have usually not been detected by the test.

749   Incidents caused due to bad Rollout Schedule or descriptions are rare. This suggests that the

750   evaluation of incidents as such is more a measure of the test quality than of the Rollout

751   Planning and execution performance.

752   One of the challenges these workshops face is related to the proper assignment of incidents to

753   the Rollout Change. The application-specific increase of 2nd level incidents after rollout provides

754   a good indication of the expected number of incidents required for analysis.

755   The second KPI introduced is specifically designed to measure the planning and execution

756   performance of a rollout. Here, the scheduled time is compared with the actual execution time.

757   The measurement can easily be conducted by the inclusion of communication items within the

758   Rollout Schedule, such as e-mails. These e-mails are sent by the rollout personnel shortly prior

759   to execution of critical items, here called 'measurement points'. The time stamps of these e-

760   mails provide a measure of the actual execution time. **Figure 13** illustrates the discrepancy

761   between the scheduled time (open circles) and actual execution time (closed circles). If the

762   closed circles are below the open ones then the actual rollout was executed quicker than

763   foreseen. If vice versa, a delay occurred. **Figure 13** illustrates that the activity related to

764   measurement point 2 finished sooner than foreseen. The activity related to measurement point

765   3, however, was delayed. Such a diagram provides a good basis for a structured discussion on

766   why certain rollout activities started later or finished earlier.

767   **5. Conclusions**

768   Within this publication a deployment framework is presented which illustrates how good ITIL

769   management practices can be aligned with a project-management model in order to assure

770   timely delivery of what this paper refers to as 'project-specific cut-over'. This alignment is

771   shown by the use of a semi-agile release model. Due to the release model selected, which is

772   particularly characterized by its bundling of projects for what is here referred to as 'release-

773   specific rollout', a new definition and interpretation of ITIL's generic view of deployment is

774   required. The facilitated release model additionally required two types of go-live scenarios: one

775   for each participating project and one for each release.

776   In order to finalize a project-specific cut-over punctually, a variety of cut-over related

777   deliverables are presented and chronologically assigned to a typical IT project managed in a

778   cascading manner. Projects already need to take into account cut-over related deliverables

779   during the initial planning phase. It is also shown how project-specific outcomes interrelate

780   with release-specific planning activities and are finally aggregated within the Rollout Handbook

781   and proposed as Rollout Change for approval by the CAB. Emphasis is placed on the Lessons-

782   learned Workshop recommended to be conducted a couple of weeks after the rollout, once the

783   level of incidents has normalised. This Workshop should be guided by the rollout/cut-over

784   related incidents, and a comparison of the scheduled time and actual execution time of the

785   rollout should be included in the Workshop.

786   Even though consulting methodologies such as ASAP (Accelerated SAP) recommend scattered,

787   project-specific deliverables useful for Cut-over Planning, this publication offers an integrated

788   approach on how to systematically prepare for a project-specific cut-over with all required

789   deliverables. The framework provided extends the ITIL release and deployment process by

790   means of IT projects and allows IT projects to interface easily with the ITIL change-management

791   process.

792   The framework has proven to be successful since it forces projects to think already during the

793   planning phase about cut-over related topics. After implementation of this framework we see

794   projects better prepared prior entering the release. However, we also do have projects which

795   wish to cut-over by the use of their own methodology. Probably this formal approach is not

796   everybody's cup of tea.

797   **Acknowledgments**

798   I would especially like to thank Andreas Veldmann for the constantly inspiring discussions and

799   his substantial help in the implementation of this framework. Thanks are also due to Necla

800   Demirci, Gerd Heyken and Joachim Kunze for many stimulating discussions on this paper and

801   related work. I am particularly grateful to Sebastian Evert, Dr Sebastian Diedrich and Enrico

802   Voigt for providing incident-related data from Cherwell's IT Service Management Software. I am

803   very grateful to Jim Blake at World2World Hamburg for editing the English version of this

804   paper.

805

References

AXELOS (2011). 'ITIL® Glossary of Terms English v. 1.0', published by © AXELOS Limited,
available at:
https://www.axelos.com/Corporate/media/Files/Glossaries/ITIL_2011_Glossary_GB-v1-0.pdf
(accessed 9th of June 2015).

BSI (2008). 'BSI-Standard 100-4: Notfallmanagement (Emergency Management) Version 1.0',
published by the German Federal Office for Information Security (Bundesamt für Sicherheit in
der Informationstechnologie).
https://www.bsi.bund.de/cae/servlet/contentblob/471456/publicationFile/30746/standard_10
04.pdf (accessed 9th of June 2015).

Cabinet Office (2011). 'ITIL Continual Service Improvement 2011 Edition (Best Management
Practices)', published by The Stationery Office (July 29, 2011), ISBN-10: 011331308X, ISBN-13:
978-0113313082.

Jäntti, M.; Virkanen, H.; Mykkänen, J.; Hotti, V. (2014). 'Exploring the Role of IT Service
Management and IT Service Governance within IT Governance.' Service Systems and Service
management (ICSSM), 2014, 11th International Conference on, IEEE Conference Publications,
pp. 1-6, DOI: 10.1109/ICSSSM.2014.6874122.

Jokela, K.; Jäntti, M. (2012). 'Challenges and Problems in Product Portfolio Release and
Deployment Management: A Case Study.', Service Systems and Service management (ICSSM),
2012 9th International Conference on, IEEE Conference Publications, pp. 138-141, DOI:
10.1109/ICSSSM.2012.6252208.

Lahtela A., Jaentti M. (2011). 'Challenges and Problems in Release Management Process: A Case
Study.' Software Engineering and Servcie Science (ICSESS) 2011 IEEE, 2nd International
Conference on, IEEE Conference Publications, pp. 10-13, DOI: 10.1109/ICSESS.2011.5982242.

833   Larsson, F.; Rusu, L.; Aasi, P. (2015). 'Organizational Structure in IT Governance: A Case Study of

834   an IT Governance governance Implementation Project.' AMCIS 2015 Proceedings 06/2015, 21st

835   Americas Conference on Information Systems, Puerto Rico 2015,

836   http://aisel.aisnet.org/amcis2015/SocTech/GeneralPresentations/16/ (accessed 30th of

837   September 2015).

838

839   Marrone, M.; Gacenga, F.; Cater-Steel, A.; Kolbe, L. (2014). 'IT Service Management: A Cross-

840   national Study of ITIL Adoption.' Communications of the Association for Information Systems,

841   Volume 34, Article 49, http://aisel.aisnet.org/cgi/viewcontent.cgi?article=3772&context=cais

842   (accessed 30th of September 2015).

843

844   Nageldinger, G. (2014). 'Project-specific cut-over and release-specific rollout: what is the

845   difference and what should we look out for?' ('Projekt-spezifischer Cut-over und Release-

846   spezifischer Rollout: Was ist der Unterschied? Worauf müssen wir achten?') – a paper

847   presented at the 5th Annual Convention on IT Change and Release Management ('5.

848   Jahrestagung IT Change und Release Management'), a marcus evans event, 26-27.6. 2014,

849   Berlin, Germany.

850   Nageldinger, G. (2015). 'IT's missing chapter: interrelations and interdependencies between

851   deployment and project management' ('ITILs fehlendes Kapitel: Die Wechselwirkung zwischen

852   dem Deployment – und Projektmanagement') – a paper presented at iqnite Europe 2015, 28-

853   30.4.2015, Düsseldorf, Germany; abstract available at http://www.iqnite-

854   conferences.com/de/programm/abstracts/nageldinger_ab.pdf (accessed 13.7.2015).

855   PMI (Project Management Institute) (2001). 'A guide to the project management body of

856   knowledge (PMBOK® guide) – 2000ed.' ISBN: 1-880410-23-0.

857   Proehl, T.; Erek, K.; Limbach, F.; Zarnekow, R. (2013). 'Topics and Applied Theories in IT Service

858   Management.' System Sciences (HICSS), 2013, 46th Hawaii International Conference on, IEEE

859   Conference Publications, pp. 1367-1375, DOI: 10.1109/HICSS.2013.555.

860

861   Rance, S. (2011). 'ITIL Service Transition 2011 Edition (Best Management Practices)', published

862   by The Stationery Office (July 29, 2011), ISBN-10: 0113313063, ISBN-13: 978-0113313068.

863   SAP AG (2009). 'Template Management: Using Templates in Global Rollout, Solution

864   Management - Application Lifecycle Management', available at:

865   http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/20c957bc-01bb-2c10-

866   0bb1-ed8c458b5b09?QuickLink=index&overridelayout=true&45896020704110 (accessed 16th

867   of June 2015).

868   Shahsavarani, N.; Ji, S. (2011). 'Research in Information Technology Service Management (ITSM):

869   Theoretical Foundation and Research Topic Perspectives.' CONFIRM 2011 Proceedings (30), 2011,

870

871   Shivashankarappa, A.N.; Dharmalingam, R.; Smalov, L.; Anbazhagan, N. (2012). 'Implementing it

872   Governance Using Cobit: A Case Study Focusing on Critical Success Factors' Internet Security (WorldCIS),

873   2012 World Congress on, pp. 144-149, ISBN: 978-1-4673-1108-3,

874   http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6280217&url=http%3A%2F%2Fieeexplo

875   re.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6280217 (accessed 16th of June 2015).

876

877   Tanovic, A.; Orucevic, F. (2011). 'Integration of PRINCE2 model into ITIL V3 model.' Telecommunications

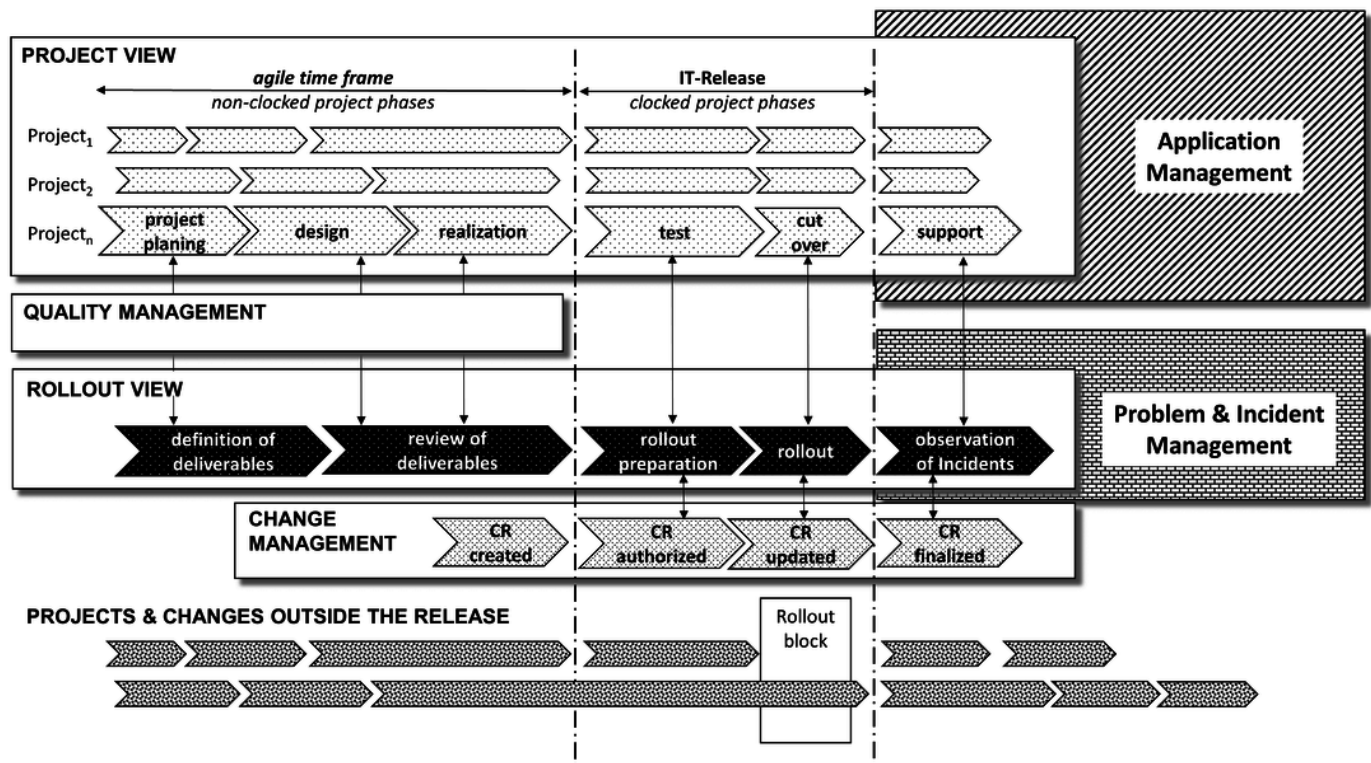
878   Forum (TELFOR), 2011, 19th, IEEE Conference Publications, pp. 102-105, DOI:

879   10.1109/TELFOR.2011.6143503

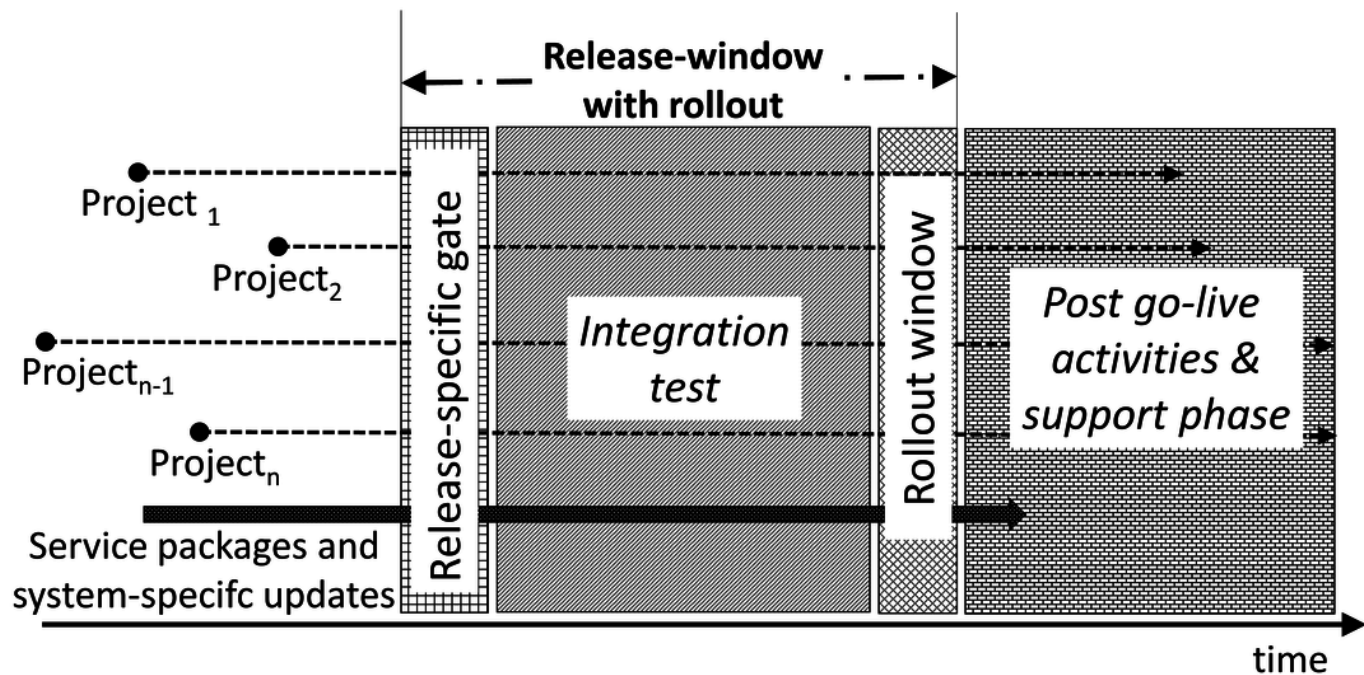
880

1

Example of a semi-agile release model

Example of a semi-agile release model with its project and rollout view and relation to ITIL's
Application Management, Problem & Incident Management as well as Change Management

2

release phase and rollout window

Illustration of the release phase and the positioning of the rollout window

3

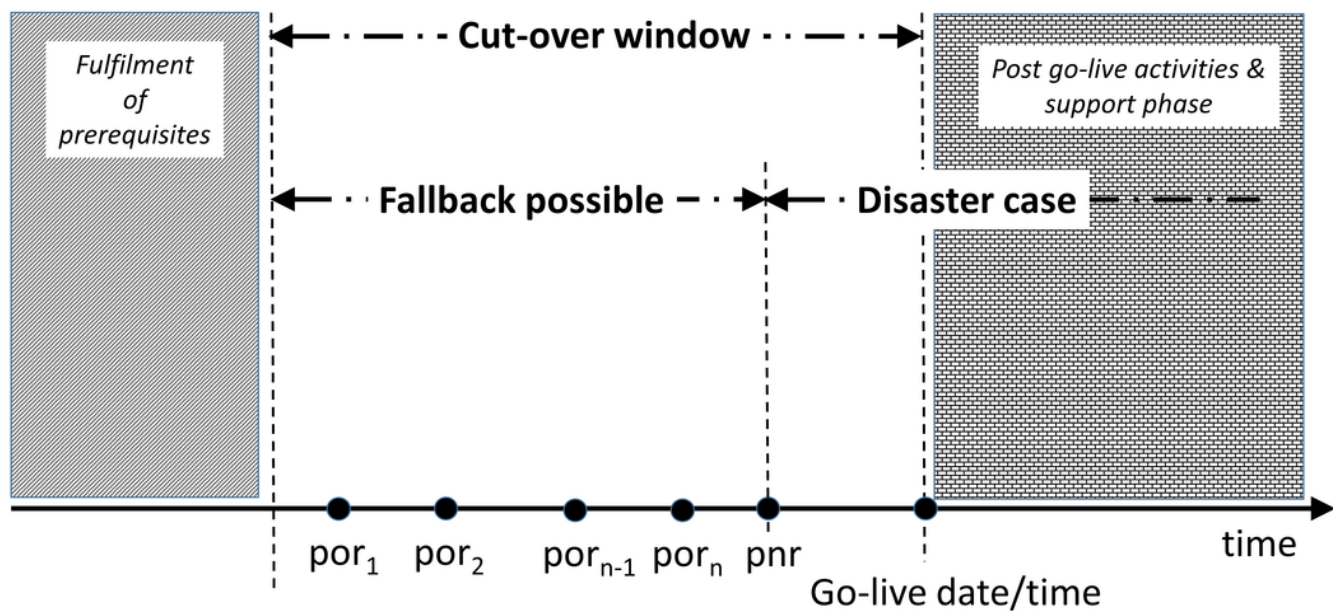
Visualization of the cut-over window

4

Illustration of two release-specific rollouts.

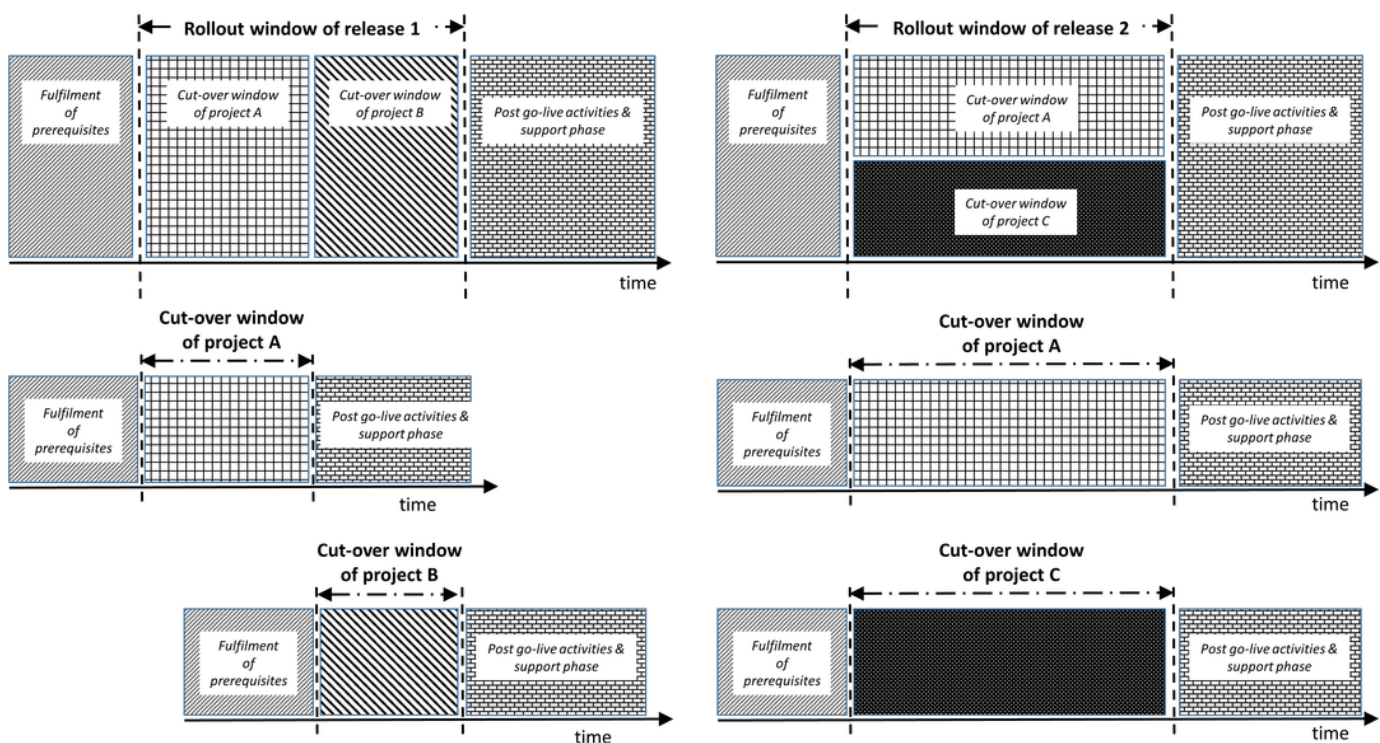
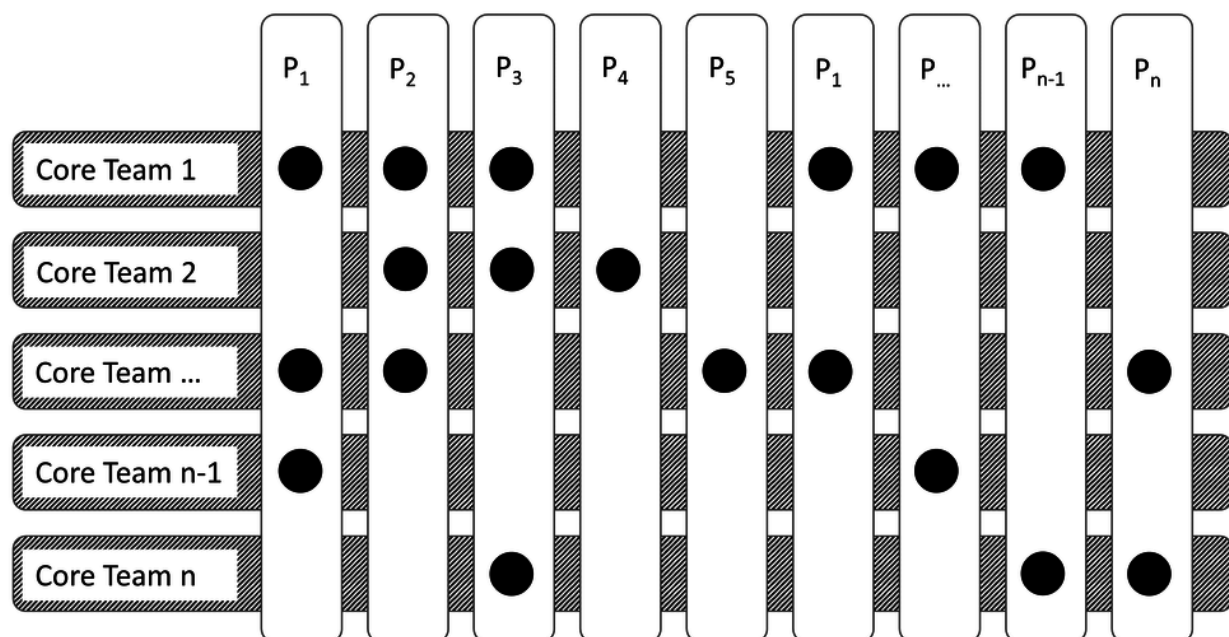
Illustration of two release-specific rollouts. The Go-live Strategy of project A utilizes the Rollout Window of release 1 and 2, whereas projects B and C facilitate only one Rollout Window. In the same way as the project's Go-live Strategy, the release-specific rollout uses a Go-live Strategy for the release. In this illustration, the rollout of release 1 utilizes a sequential rollout strategy and the rollout of release 2 a parallel rollout strategy.

Dependency Matrix

Illustration of a Dependency Matrix, which is used as a tool for the evaluation of the
(potential) impact and interrelations between Project Teams and Core Teams; Each bold
point symbolizes an interaction between a Project Team and a Core Team; the sum of
interactions is a criterion for the complexity of a project as well as a release.

6

planning activities of the release

Schedule of planning activities of the release, associated with the publication time of rollout newsletter

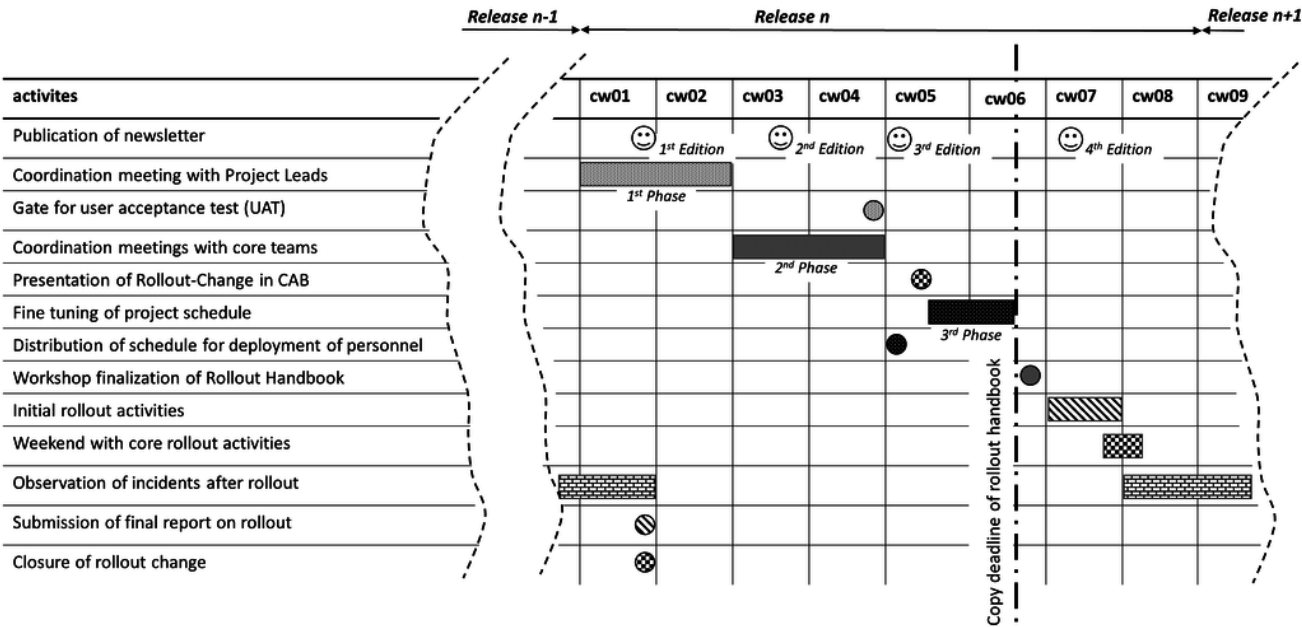



Illustration of incidents during a two-year period

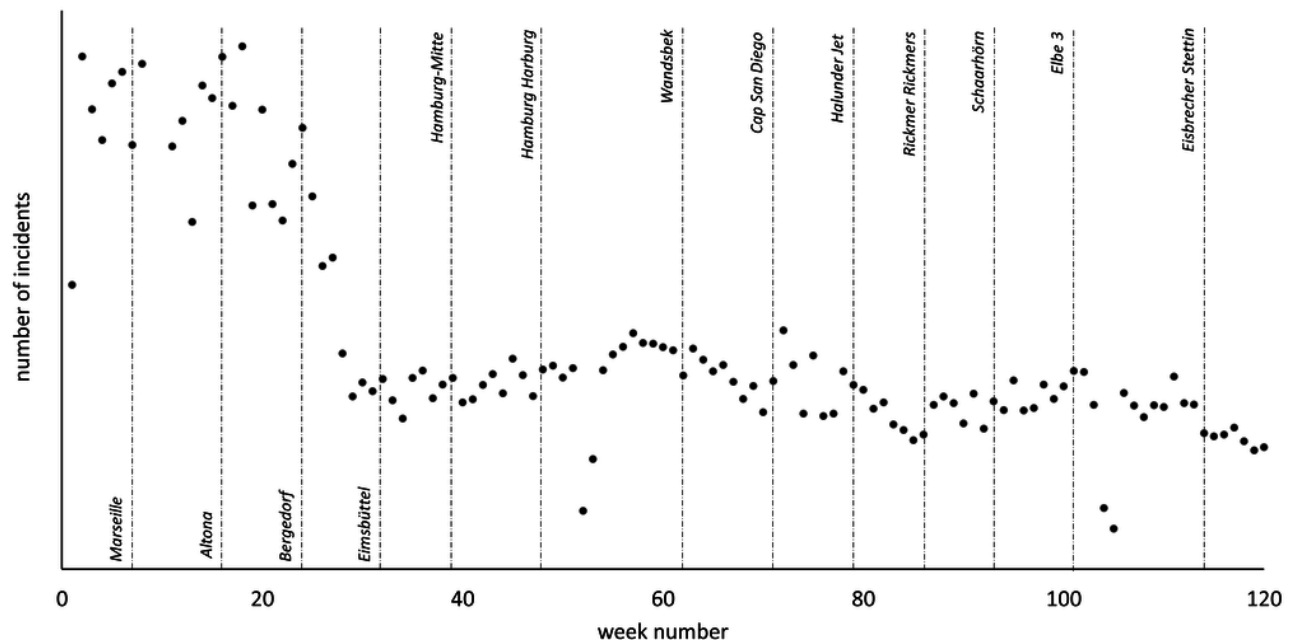
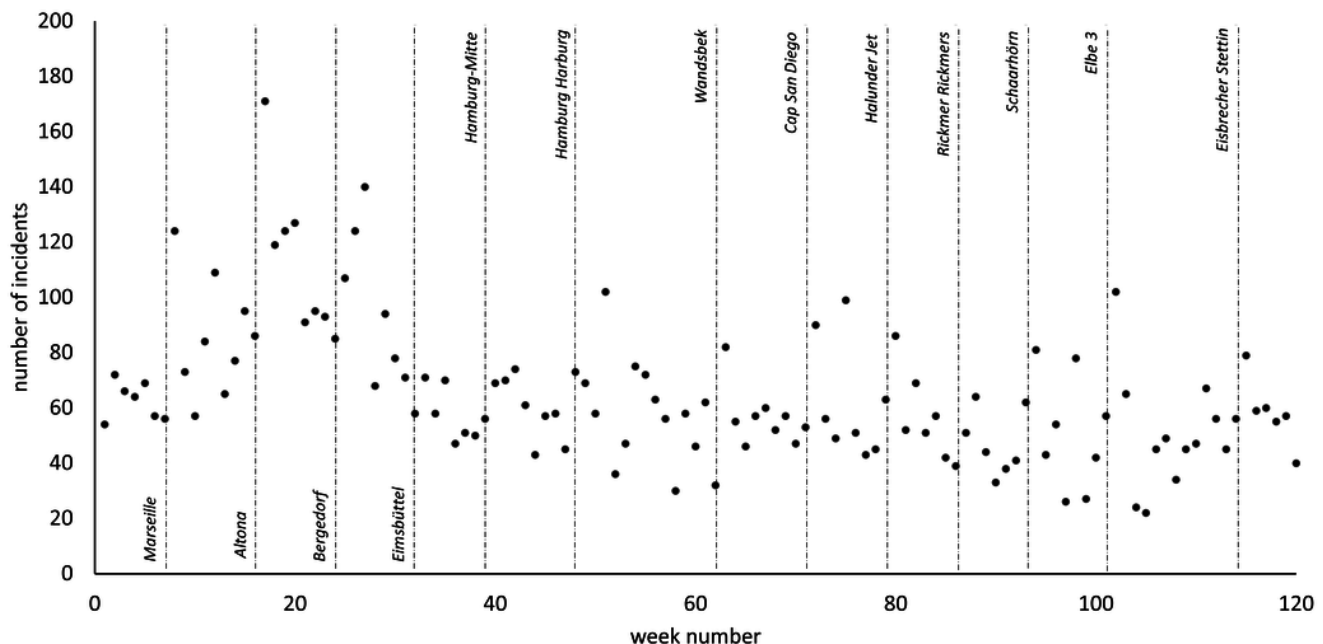
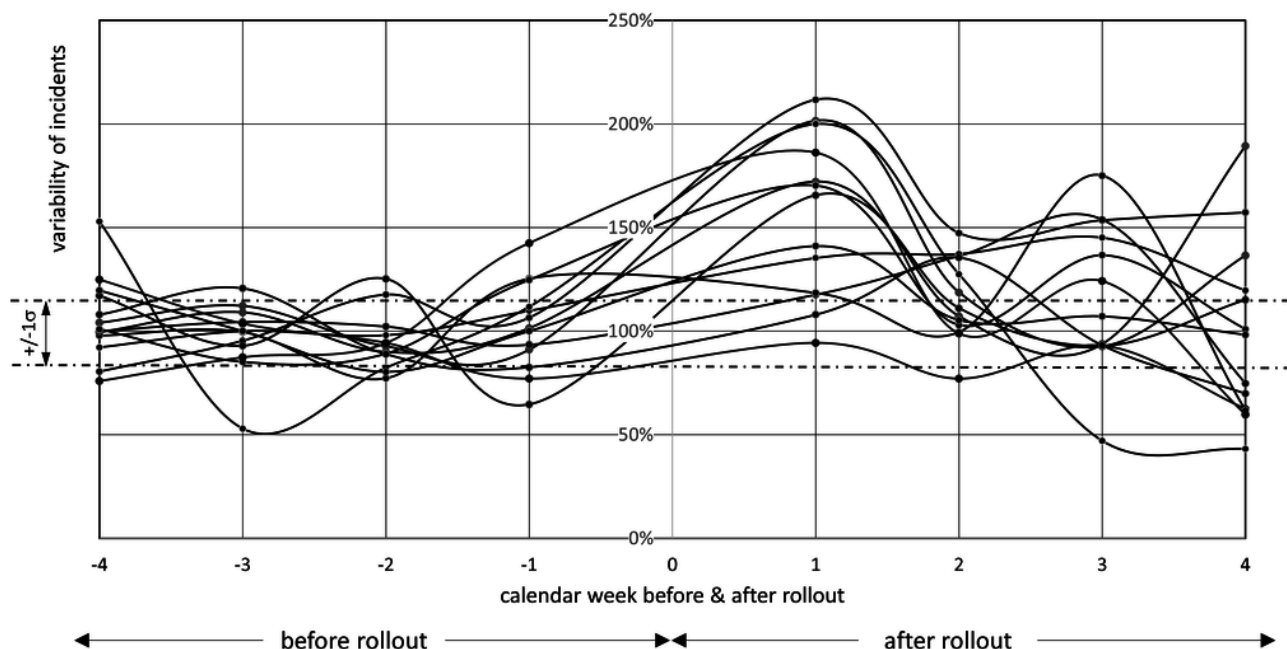
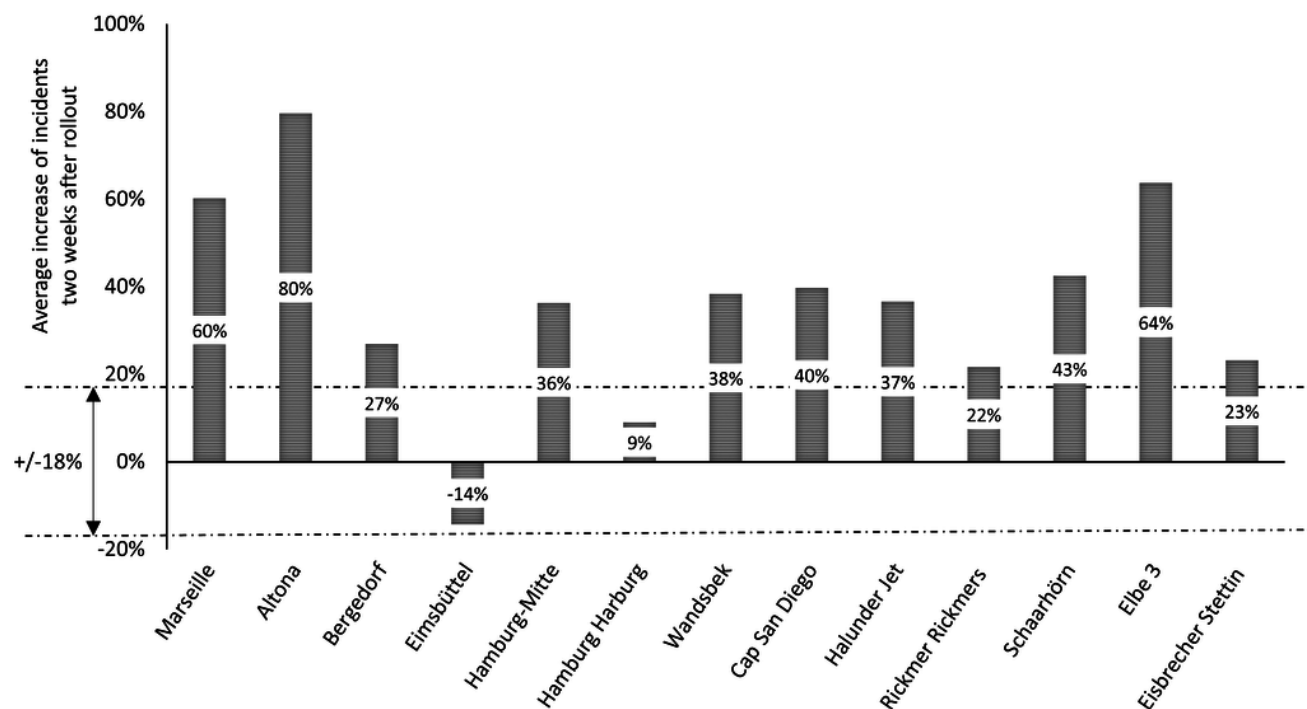
Illustration of incidents during a two-year period. Dotted lines illustrate week of rollout with release names; data relate to the total number of incidents recorded. Actual incident numbers are disclosed due to confidentiality reasons.

second-level incidents of a main order-handling system

Illustration of incidents during a two-year period. Dotted lines illustrate week of rollout with release names; data relate to second-level incidents of a main order-handling system, which participates in all rollouts.

9

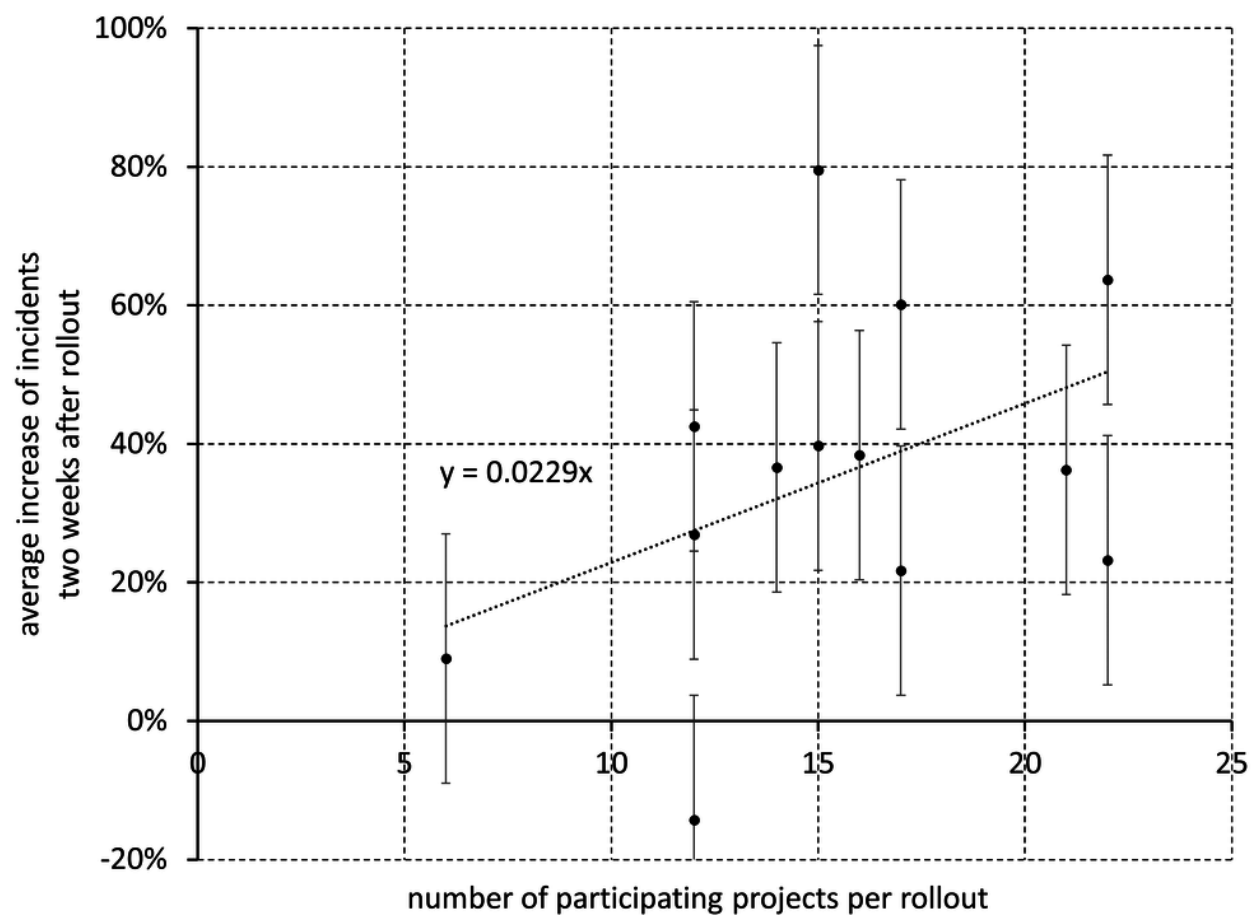
Variability of incidents four weeks before and after rollout.

Variability of incidents four weeks before and after rollout. Connecting lines visualize the form of the curves; the mean value of incidents during a period of four weeks before rollout is set to 100%, with a standard deviation of +/- 18%

10

Average increase of incidents two weeks after rollout

Average increase of incidents two weeks after rollout. Standard deviation is +/- 18% as illustrated in Figure 9; names label the rollouts, which are plotted on a timeline in Figure 8.

11

Correlation coefficient A

Correlation coefficient of 0.4 between the average increase in incidents two weeks after rollout and the number of participating projects per release

Correlation coefficient B

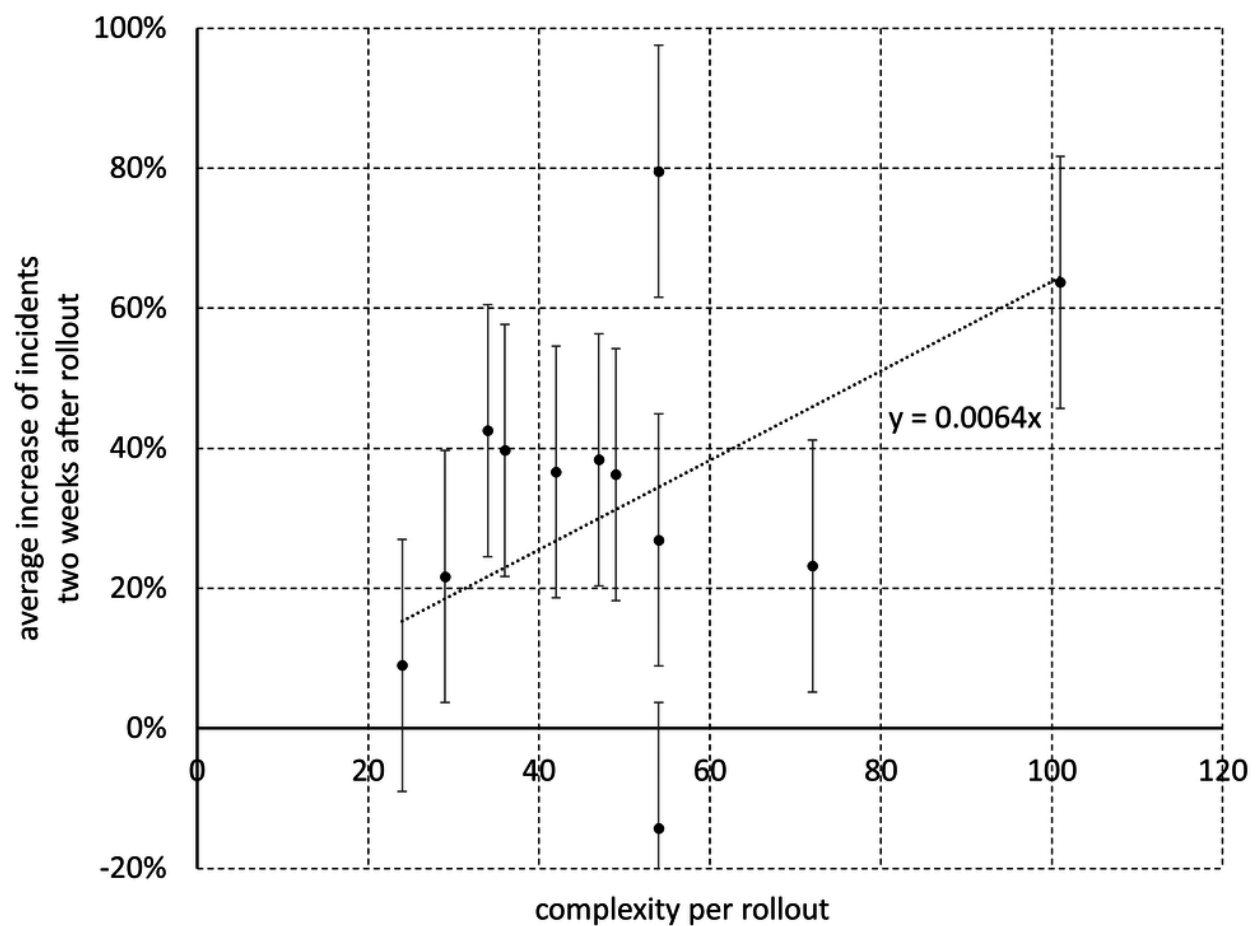
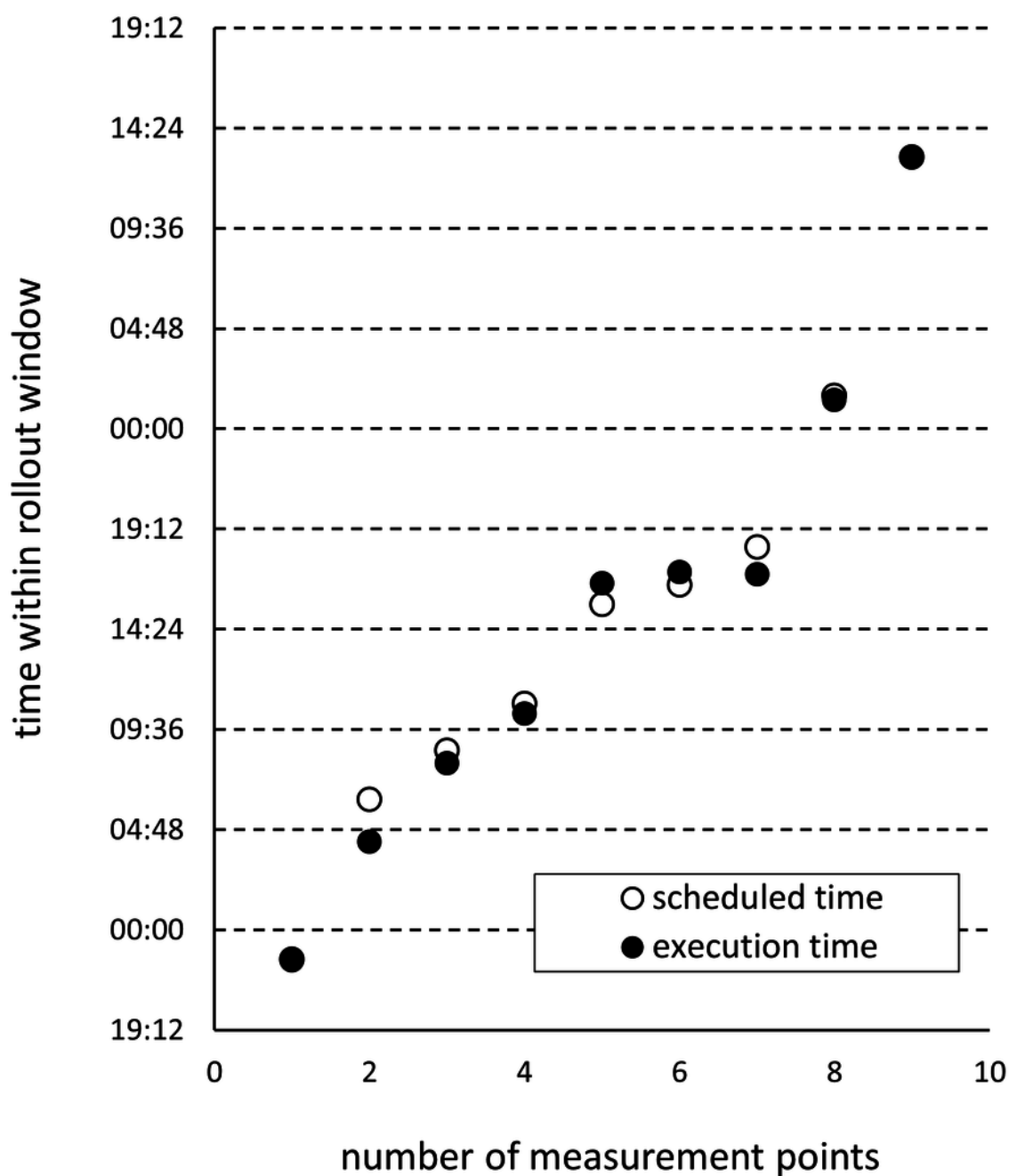
Correlation coefficient of 0.36 between average increase of incidents two weeks after rollout and the complexity per rollout

discrepancy between the scheduled time and actually executed time

Illustration of the discrepancy between the scheduled time and actually executed time for nine measurement points

Project-specific deliverables for cut-over planning.

Project-specific deliverables for cut-over planning. Bold font used for deliverables related to Rollout Management; italic font used for optional deliverables which are not evaluated by Quality Management; normal font used for mandatory deliverables.

Project planning	Design	Realization	Test	Cut-over	Support
Plan for Deployment Management	*System-profile description*	*Decision model presentation for the Go-live & Fallback Strategy*	FINAL Cut-over Plan	Operational Cut-over	Observation of incidents
Establish Cut-over Manager (COM) position	System landscape design (SLDe)	Decision related to the Go-live & Fallback Strategy	Fallback concept		Lessons learned
Initial dialogue with Rollout Management	Extension to SLDe due to Go-live Strategy	One-Pager	Disaster-recovery concept		
	Go-live strategy	DRAFT Cut-over Plan	**Plan for the deployment of personnel**		
	Fallback Strategy	Risk Register	Cut-over test		
	Framework for cut-over planning	Dependency Matrix	*Go-live simulation*		
	Stakeholder analysis	**Dialogue with Rollout Management**			
	Kick-off				
	Communication Plan				
	Completely defined Cut-over Team				

Activitytypes used for a Rollout/Cut-over Schedule. Ordinary activities are notassigned an activity type.

Activitytypes used for a Rollout/Cut-over Schedule. Ordinary activities are notassigned an activity type.

Activity type	Explanation
info	Someone needs to be informed, usually by e-mail
checkpoint	Technical verification point within cut-over/rollout flow
optional	Activity is conducted only for certain rollouts; if the activity is not required then the time is set to 0
security	Security-related activity, such as back-up or the implementation of restore points
handover	Handover of a certain document, protocol etc.
unique	Non-recurring activity within the rollout flow; commonly relates to a project and is therefore not facilitated for other rollouts
breakpoints	Breakpoints are used during task or process chains in order to verify intermediate results. This activity type relates to these breakpoints.
comment	Project-planning tools often provide limited functionality related to comments within the working breakdown structure (WBS). Comments are activities with time=0 and are merely used to provide additional information within the WBS.

Incidentcategories used solely for rollout evaluation.

Incidentcategories used solely for rollout evaluation. This classification scheme isused in addition to the schema already facilitated by the incident managementprocess, which is not further described here.

Incident Category	Description
Error within release/ not detected through testing	Incident was caused by a software error which was not detected in testing. This category is also used for service requests, which fix release-specific errors, e.g. in the form of patches.
Human failure	This category is used for incidents caused by human failure, such as faulty execution of patches or noncompliance with agreements.
Incident caused by project	Incident is related to a project; this category makes particular sense if a significant number of incidents are related to a particular project.
Rollout Schedule requires change	Activity required to be conducted was not properly described. If it is a reoccurring activity then the description needs to change.
Expected behaviour	Irregularities recorded are considered to be an accepted and expected behaviour of the system or application.
Sequential error within the Rollout Schedule (application-specific)	Incident is caused by a sequential error within the Rollout Schedule; activity was requested to be conducted by an application which regularly participates during the rollout.
Sequential error within the Rollout Schedule (project-specific)	Incident is caused by a sequential error within the Rollout Schedule; the activity was requested to be conducted by a Project Team.
Bugfix	This category relates to incident-response measures which occur prior to the rollout and which aim to eliminate software error. It is merely a convention to relate them to the Rollout Change, which also fixes these errors.
Missing contact person	The contact person who logged this change could not be contacted. The number of incidents assigned to this category should be kept small.
Incident is not reproducible	The incident cannot be reproduced.
Wrongly connected	Incident was not related to the Rollout Change and was wrongly related.
Not classifiable	Incident cannot be classified with available classification; the classification is of course extended with additional categories where required.