CHANGEGROUP

Analysing Rabobank Group ICT

Analysing ITIL data to provide fact based predictive model of workload to Service Desk from the Software Change Process

International Business Process Intelligence 2014 Competition

Author:	John Hansen, ChangeGroup Partner
Date:	1st July 2014
Address:	ChangeGroup Flæsketorvet 68 DK-1711 Copenhagen K Denmark

Table of Contents

Introduction

The purpose of this document is to answer the questions raised by Rabobank Group ICT in the International Business Process Intelligence 2014 competition. The scope of this analysis is the questions raised (described in the "Subquestions To Be Answered" section in this document). The answers all contain assumptions. Those assumptions might be incorrect/challenged – but the analysis can easily be re-run on other assumptions.

Tools Used

In order to solve the case the following tools has been user:

- Disco 1.6.7 (Fluxicon)
- Excel (Microsoft Excel for Mac 2011)
- Word (Microsoft Word for Mac 2011)

Approach

The questions raised has been analysed and answered using the following approach:

1. Importing log files into Disco
2. Auto-generating As-Is processes using Disco
3. Getting to know the process – analysing As-Is processes in Disco
4. Performing business process mining and discovery using Disco analysis and Excel

Subquestions To Be Answered

The following Subquestions are the scope of this analysis, and will be analysed and answered separately in the sections following this section.

Each subquestion will be answered in a section in this document, where the section will be named "Subquestion X – Title" where X refers to the subquestion number and "title" refers to the subquestion title – for example "Subquestion 1 - Identification of Impact-patterns".

Subquestions to be answered:

1. **Identification of Impact-patterns**: We expect there to be a correlation between the implementation of a change and the workload in the Service Desk (SD) and/or IT Operations (ITO), i.e. increased/decreased volume of Closed Interactions and/or increased/decreased volume of Closed Incidents. Rabobank Group ICT is interested in identifying any patterns that may be visible in the log for various service components to which a configuration item is related, in order to predict the workload at the SD and/or ITO after future changes.
2. **Parameters for every Impact-pattern**: In order to be able to use the results of prior changes to predict the workload for the Service Desk directly after the implementation of future changes, we are interested in the following parameters for every impact-pattern investigated in sub question 1:
   a. What is the average period to return to a steady state?
   b. What is the average increase/decrease of Closed Interactions once a new steady state is reached?
3. **Change in Average Steps to Resolution**: Since project managers are expected to deliver the same or better service levels after each change implementation, Rabobank Group ICT is looking for confirmation that this challenge is indeed being met for all or many Service Components.

4. **Creativity challenge**: Finally, we challenge the creative minds, to surprise Rabobank Group ICT with new insights on the provided data to help change implementation teams to continuously improve their Standard Operation Procedures.

Answers To Analysis Questions

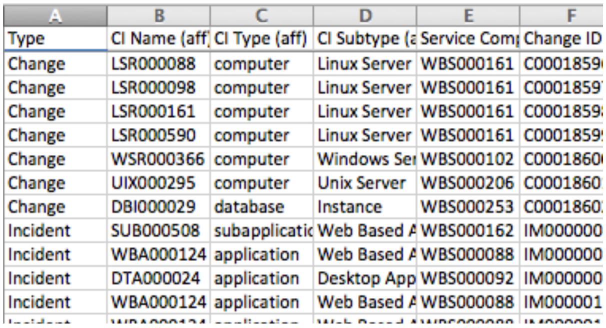
In this section the four subquestions will be analysed and answered separately.

Subquestion 1 - Identification of Impact-patterns

In order to answer the question in subquestion 1, which is analysing the correlation between the implementation of a change and the workload in the Service Desk (SD) and/or IT Operations (ITO), i.e. **increased/decreased volume** of **Closed** Interactions and/or increased/decreased volume of Closed Incidents it is needed to determine the assumptions under which the analysis is performed:

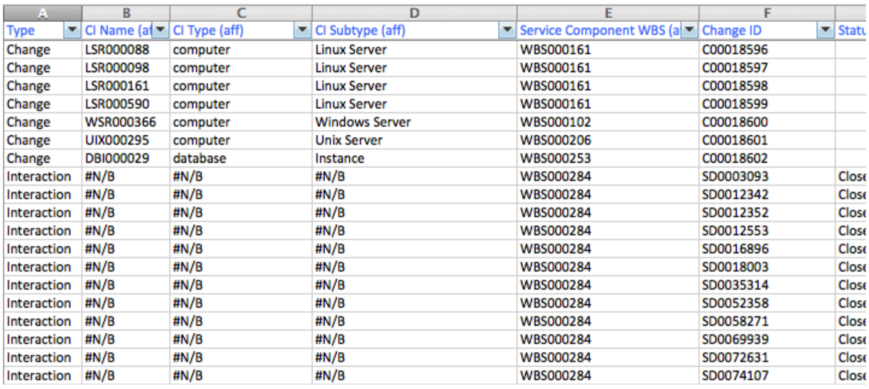
Assumptions:

1. It is assumed that the question is best answered by counting Interactions/Incidents before a Change, and compare that number to the number of Interactions/Incidents after a Change. In order to do so the Changes log file and the Incidents/Interactions log files has been concatenated together (resulting in two files: Changes+Interactions and Changes+Incidents) and then imported into Disco. Type (Interaction, Incident or Change) is added to each event.



A	B	C	D	E	F
Type	CI Name (aff	CI Type (aff)	CI Subtype (a	Service Com	Change ID
Change	LSR000088	computer	Linux Server	WBS000161	C0001859
Change	LSR000098	computer	Linux Server	WBS000161	C0001859
Change	LSR000161	computer	Linux Server	WBS000161	C0001859
Change	LSR000590	computer	Linux Server	WBS000161	C0001859
Change	WSR000366	computer	Windows Se	WBS000102	C0001860
Change	UIX000295	computer	Unix Server	WBS000206	C0001860
Change	DBI000029	database	Instance	WBS000253	C0001860
Incident	SUB000508	subapplicati	Web Based A	WBS000162	IM000000
Incident	WBA000124	application	Web Based A	WBS000088	IM000000
Incident	DTA000024	application	Desktop App	WBS000092	IM000000
Incident	WBA000124	application	Web Based A	WBS000088	IM000001

Figure 1 - Example Of Changes Concatenated With Incidents



A	B	C	D	E	F	
Type	CI Name (af	CI Type (aff)	CI Subtype (aff)	Service Component WBS (a	Change ID	Statu
Change	LSR000088	computer	Linux Server	WBS000161	C00018596	
Change	LSR000098	computer	Linux Server	WBS000161	C00018597	
Change	LSR000161	computer	Linux Server	WBS000161	C00018598	
Change	LSR000590	computer	Linux Server	WBS000161	C00018599	
Change	WSR000366	computer	Windows Server	WBS000102	C00018600	
Change	UIX000295	computer	Unix Server	WBS000206	C00018601	
Change	DBI000029	database	Instance	WBS000253	C00018602	
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0003093	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0012342	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0012352	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0012553	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0016896	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0018003	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0035314	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0052358	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0058271	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0069939	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0072631	Clos
Interaction	#N/B	#N/B	#N/B	WBS000284	SD0074107	Clos

Figure 2 - Example Of Changes Concatenated With Interactions

2. It is assumed that Changes timestamp “Change Record Open Time” and “Change Record Close Time” are system generated and actual times and should be used for concatenating with Incidents timestamp “Open Time” and “Close Time”
3. It is assumed that Changes timestamp “Change Record Open Time” and “Change Record Close Time” are system generated and actual times and should be used for concatenating with Interactions timestamp “Open Time (First Touch)” and “Close Time”
4. It is assumed that the term “Implementing Changes” is best represented as the collection of Changes, because a single Change followed by another Change two days later will be considered one major Change by the users – thus the Interactions and Incidents placed after the Changes will be impossible to distinguish if belonging to the first or second Change. It is assumed that “A Change” is the collection of Changes (in Figure 3 the collection of the three Changes would constitute “A Change”[1] – Figure 4 shown the same concept for Incidents.)
5. It is assumed that the correlation should be analysed for Interactions/Incidents before a collection of Changes and Interactions/Incidents after a collection of Changes as illustrated in Figure 3 and Figure 4. As it can be seen Interactions/Incidents placed within the collection of Changes are belonging to both before and after Changes in order to make them neutral.

[1] A drawback is that Changes with long time between the mare still considered one Change. It is assumed this is not destroying the overall picture. Furthermore other perspectives are also made on the analysis of correlations

[2] Of course other assumptions regarding period in time a Change affects Interactions closed could be measured. The dataset could be manipulated to

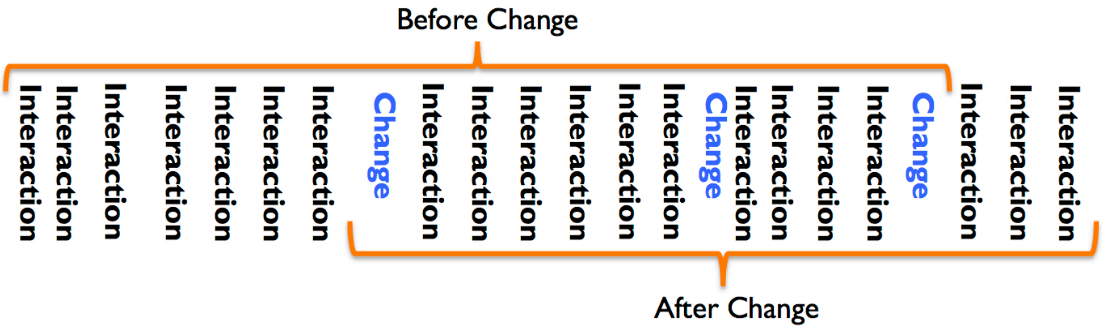


Figure 3 - Interactions Before And After Changes For A Given Product

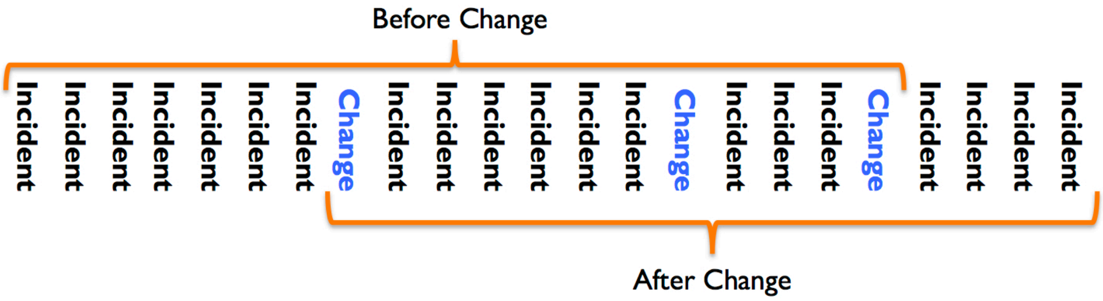


Figure 4 - Incidents Before And After Changes For A Given Product

6. It is assumed that Counting Interactions before Changes is correct according to what is illustrated in Figure 3 and Figure 5 and that counting Interactions after Changes is correct according to what is illustrated in Figure 3 and Figure 6
7. It is assumed that Counting Incidents before Changes is correct according to what is illustrated in Figure 4 and Figure 7 and that counting Incidents after Changes is correct according to what is illustrated in Figure 4 and Figure 8

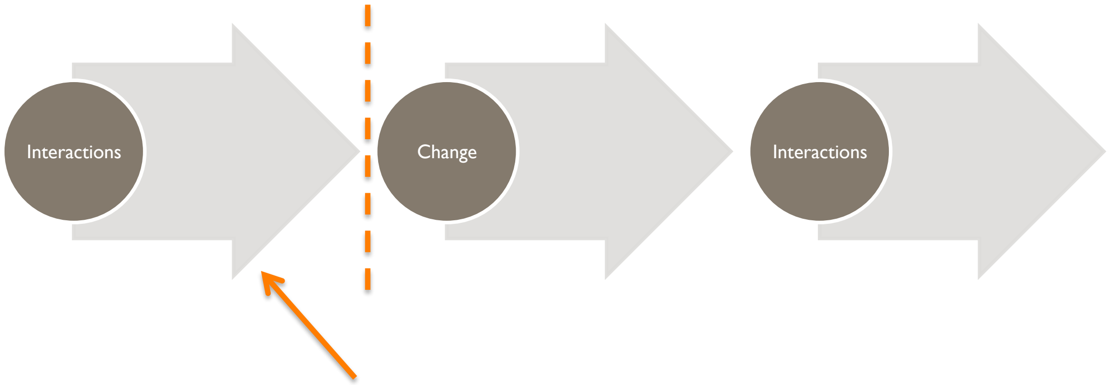


Figure 5 - Counting Interactions Before A Related Change Occur

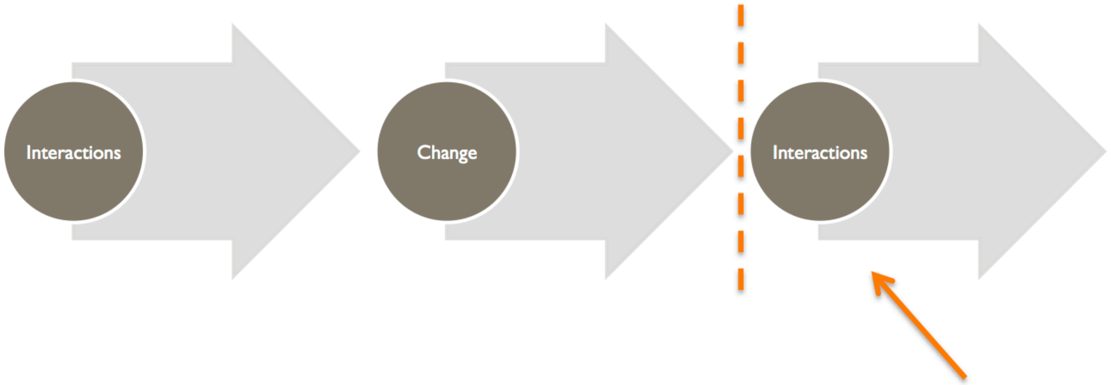


Figure 6 - Counting Interaction After First Occurrence of A Related Change

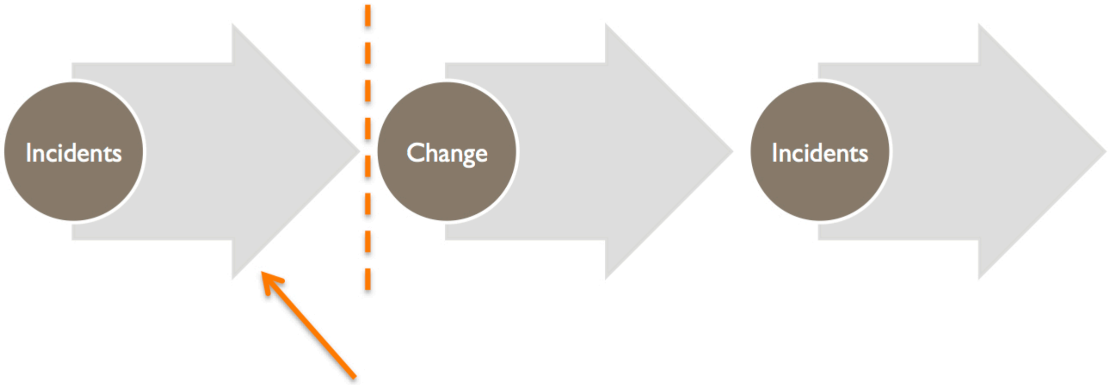


Figure 7 - Counting Incidents Before A Related Change Occur

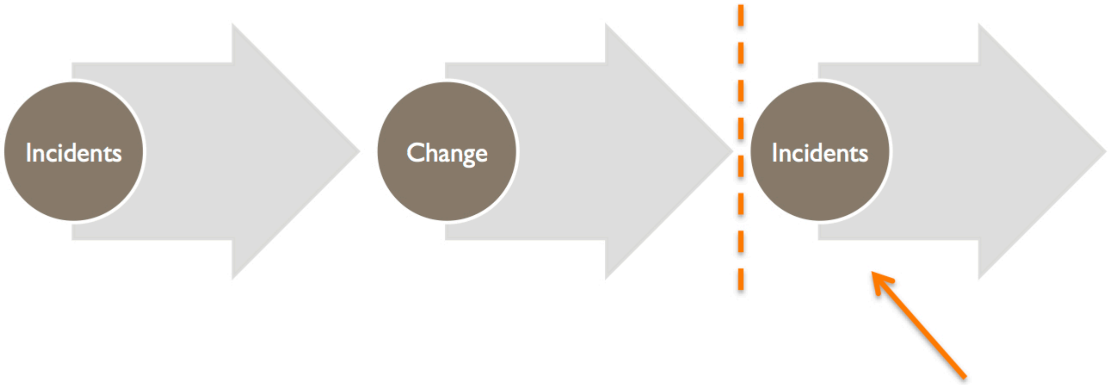


Figure 8 - Counting Incidents After A Related Change Occur

8. It is assumed that the few incomplete cases do not need to be filtered out since the number of incomplete cases are very low

Strategy For Analysing Subquestion 1

The following hypothesis are examined in order to verify, if there is indeed a correlation between implementing a Change and the following workload in Service Desk and / or IT Operations:

1. **Hypothesis Q1.1**: Is there in general a significant increase/decrease in the number of interactions after implementing Changes compared to before implementing Changes?
2. **Hypothesis Q1.2:** Is there a significant correlation between the increase in number of interactions being related to PRODUCTS with Changes, after the Changes have been implemented
3. **Hypothesis Q1.3**: Is there in general a significant increase/decrease in the number of incidents after implementing Changes compared to before implementing changes?
4. **Hypothesis Q1.4:** Is there a significant correlation between the increase/decrease in number of incidents being related to PRODUCTS with Changes, after the Changes have been implemented

Examining Hypothesis Q1.1

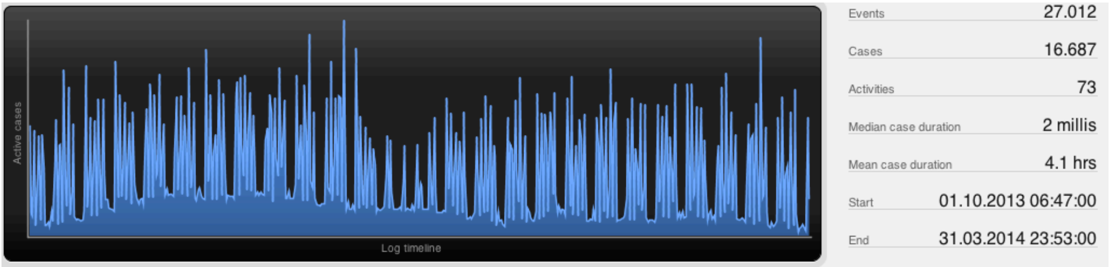
Approach (Hypothesis Q1.1)

First the dataset containing Changes is examined. All changes are commenced after 01.10.2014 06.47 (see Figure 9). This dataset is then filtered so only those changes having a timestamp in Actual End is included, in order to have closed/implemented changes only. This gives 92% of all cases in the Changes dataset. It is assumed that no changes have been implemented before 01.10.2013 06.47 – probably due to implementing the first release update at a date after 01.10.2013.

Second, the dataset containing all the Interactions is examined. In order to determine if there is a significant impact in general on the number of Interactions, after Changes have been implemented. The number of Interactions before and after implementing Changes is examined.

Result (Hypothesis Q1.1)

The filtered Changes dataset contains 92% of all cases (only the implemented ones) and the dataset starts 01.10.2013 06.47.



Events	27.012
Cases	16.687
Activities	73
Median case duration	2 millis
Mean case duration	4.1 hrs
Start	01.10.2013 06:47:00
End	31.03.2014 23:53:00

Figure 9 - Detail Change - Active Cases Over Time

The Interactions dataset is filtered (time filter to match time span for Changes) to show Interactions related to before-implementing-changes and after-implementing-changes. As it can be seen in the figure below (Figure 10), the majority of Interactions are taking place after-implementing-changes (the blue area, containing cases after 01.10.2013 06.47), which is in fact 98% of all Interactions.

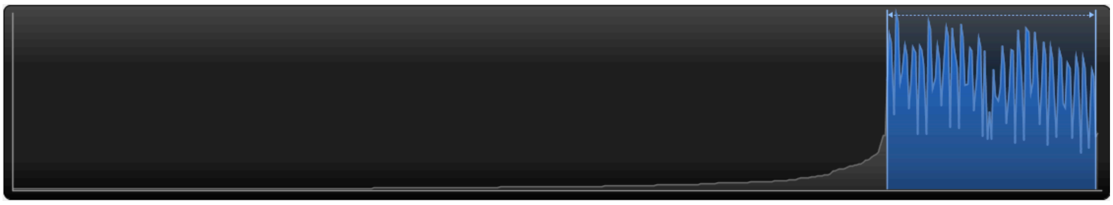


Figure 10 - Interactions Before And After Implementation of Changes

From an all-things-equal perspective there seems to be a strong correlation that implementing Changes leads to a significant increase in the number of Interactions. However, it could also be that there are other reasons to increase in Interactions than the occurrence of changes (for instance a high number of Interactions leads to more Changes). To see if there is a strong correlation between implementing Changes and the number of Interactions at the Service Desk, it is needed to look in detail into the numbers – this is done in "Examining Hypothesis Q1.2".

Examining Hypothesis Q1.2

From examining hypothesis Q1.1 it seems like there is an increase in Interactions following implementation of Changes. In this hypothesis we will look into more detail on analysing if there is a strong correlation.

Approach (Hypothesis Q1.2)

The attribute "CI Name (aff)" from the Changes dataset refers to a configuration item and is part of one or more services.

It is assumed here that by looking at the "CI Name (aff)" it can be determined which product parts (Configuration items) is being part of changes, interactions and incidents. It is also assumed that the fields mentioned in assumption 3 in the Assumptions section should be used for the cases.

Data To Be Analysed

The concatenated Changes and Interactions are imported into Disco with Type (Change or Interaction) as Activity, "CI Name (aff)" as ID, and the date fields mentioned in assumption 3 in the Assumptions Section as time stamps. This generated 1 case for each Product with Changes and Interactions as activities.

Data Filtering

In order to find the Interactions before Changes are implemented, the data (as described in the "Data To Be Analysed" are filtered so that:

- Remove cases not having first event as an Interaction event, as cases starting with a Change event is not driven by Interactions and can't compare to state before Change implementation
- Remove cases having no Change activities at all. Cases consisting of only Interactions are not interesting in this analysis
- Trim cases to contain all activities from the first Interaction event to the last Change event.
- Remove all Change activities in order to count the Interactions

In order to find the Interactions after Changes are implemented, the data (as described in the "Data To Be Analysed") are filtered so that:

- Remove cases not having first event as an Interaction event, as cases starting with a Change event is not driven by Interactions and can't compare to state before Change implementation

- Remove cases having no Change activities at all. Cases consisting of only Interactions are not interesting in this analysis
- Trim cases to contain all activities from the first Change event to the last Interaction event.
- Remove all Change activities in order to count the Interactions

Analysis

In order to determine if there is a strong correlation between implementing Changes and the number of Interactions, the following approach is applied:

1. Count the number of Interactions (counting activities as each activity represents a Closed Interaction) made for a given product before Changes are implemented for that given product (see Figure 5)
2. Count the number of Interactions (counting activities as each activity represents a Closed Interaction) made for a given product after Changes are implemented for that given product (see Figure 6)
3. Compare the number of interactions for a given product before and after implementing changes in order to see if there is a correlation between those actions.

Result (Hypothesis Q1.2)

After filtering the dataset to contain only the Interactions where the related product eventually will be subject for a change, the number of Interactions is 44758 distributed on 194 products. Since the Interaction events represent cases (1 event per Interaction) the count of Interactions is a feasible way of looking at the data from a cases perspective.

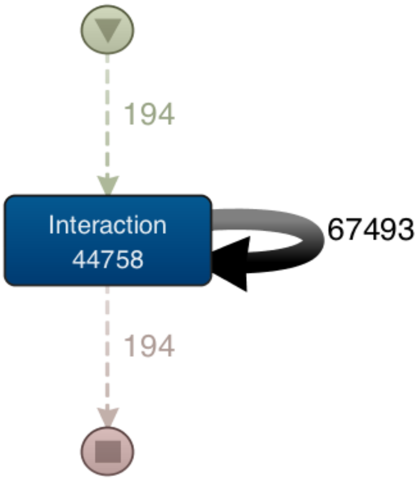


Figure 11 - Number of Interactions Before Implementing Changes For The Product The Interaction Is Related To

Now, when looking at the number of Interactions after first Change has occurred to the products (see Figure 12) they are related to, the number of Interactions increases from 44758 (see figure 11) to 50668 (see figure 12), which is an increase in the number of Interactions by 5910 (13,2%).

As it can also be seen (in Figure 11), the number of products having Interactions related to it goes down from 194 before implementing changes to 134 (in Figure 12) after implementing changes, meaning 60 (30,9 %) of the products do not have any related Interactions after changes has been implemented.

This could indicate that implementing changes means, that there is a decrease in the number on interactions to certain products while there is an increase to other products – maybe caused by introducing new issues when correcting some.

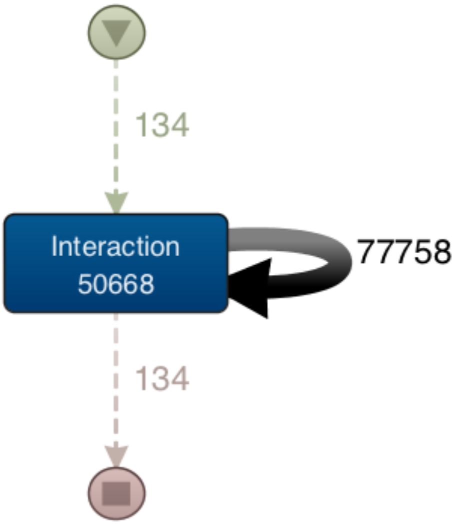


Figure 12 - Number Of Interactions After Implementing Changes For The Product The Interaction Is Related To

Another perspective on the analysis regarding the correlation between implementing Changes and the number of Interactions can be seen below.

Below (in Figure 13) are the total cases of the concatenated Changes and Interactions (Case view). Most cases with Changes are not followed by a huge number of closed Interactions. This is due to most Changes originating from Problem Management without many related Interactions prior to the Change.

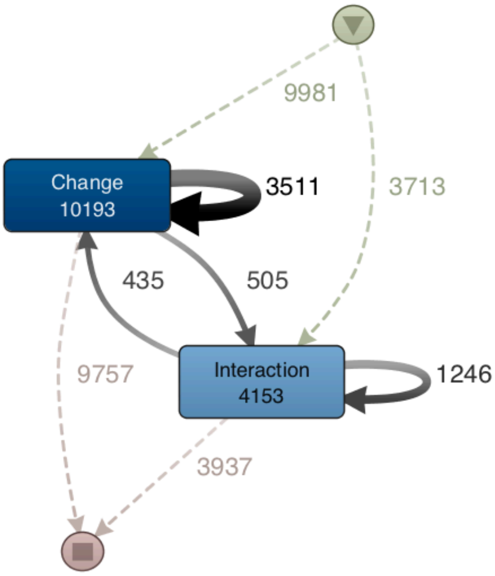


Figure 13 - Concatenated Changes And Interactions

Examining Hypothesis Q1.3

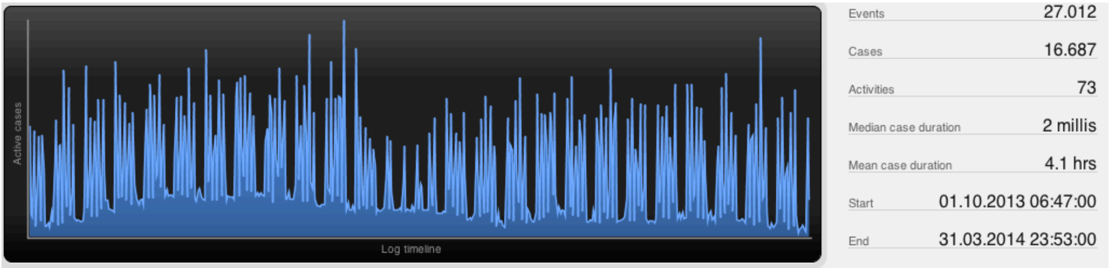
Approach (Hypothesis Q1.3)

Like in Hypothesis Q1.1 the dataset containing Changes is examined. All changes are commenced after 01.10.2013 06.47 (see Figure 15). This dataset is then filtered so only those changes having a timestamp in Actual End is included, in order to have closed/implemented Changes only. This gives 92% of all cases in the Changes dataset. It is assumed that no Changes have been implemented before 01.10.2013 06.47 – probably due to implementing the first release update at a date after 01.10.2013.

Second, the dataset containing all the Incidents is examined. In order to determine if there is a significant impact in general on the number of Incidents, after Changes have been implemented. The number of Incidents before and after implementing Changes is examined.

Result (Hypothesis Q1.3)

The filtered changes dataset contains 92% of all cases (only the implemented ones) and the dataset starts 01.10.2013 06.47.



Events	27.012
Cases	16.687
Activities	73
Median case duration	2 millis
Mean case duration	4.1 hrs
Start	01.10.2013 06:47:00
End	31.03.2014 23:53:00

Figure 14 - Detail Change - Active Cases Over Time

The Incidents dataset is filtered to show Incidents related to before-implementing-changes and after-implementing-changes. As it can be seen in the figure below, the majority of Incidents are taking place after-implementing-Changes (the blue area, containing cases after 01.10.2013 06.47), which is in fact 97% of all Incidents.

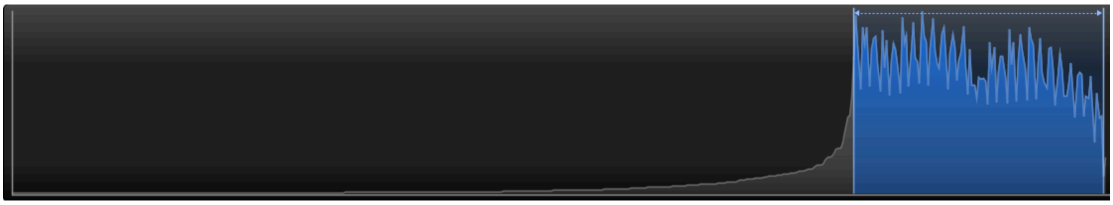


Figure 15 - Incidents Before And Aften Implementation Of Changes

Take Aways (Hypothesis Q1.3)

From an all-things-equal perspective there seems to be a strong correlation that implementing Changes leads to a significant increase in the number of Incidents. To see if there is a strong correlation between implementation of Changes and the number of Incidents at the Service Desk, it is needed to look in detail into the numbers – this is done in section "Examining Hypothesis Q1.4".

Examining Hypothesis Q1.4

From examining hypothesis Q1.3 it seems like there is an increase in Incidents following implementation of Changes. In this hypothesis we will look into more detail on analysing if there is a strong correlation.

Approach (Hypothesis Q1.4)

The attribute "CI Name (aff)" from the Changes dataset refers to a configuration item and is part of one or more services.

It is assumed here that by looking at the "CI Name (aff)" it can be determined which product parts (Configuration items) is being part of changes, interactions and incidents. It is also assumed that the fields mentioned in assumption 3 in the Assumptions section should be used for the cases.

Data To Be Analysed

The concatenated Changes and Incidents are imported into Disco with Type (Change or Interaction) as Activity, "CI Name (aff)" as ID, and the date fields mentioned in assumption 3 in the Assumptions Section as time stamps. This generated 1 case for each Product with Changes and Incidents as activities.

Data Filtering

In order to find the Incidents before Changes are implemented, the data (as described in the "Data To Be Analysed" are filtered so that:

- Remove cases not having first event as an Incident event, as cases starting with a Change event is not driven by Incidents and can't compare to state before Change implementation
- Remove cases having no Change activities at all. Cases consisting of only Incidents are not interesting in this analysis
- Trim cases to contain all activities from the first Incident event to the last Change event.
- Remove all Change activities in order to count the Incidents

In order to find the Incidents after Changes are implemented, the data (as described in the "Data To Be Analysed)" are filtered so that:

- Remove cases not having first event as an Incident event, as cases starting with a Change event is not driven by Incidents and can't compare to state before Change implementation
- Remove cases having no Change activities at all. Cases consisting of only Incidents are not interesting in this analysis
- Trim cases to contain all activities from the first Change event to the last Incident event.
- Remove all Change activities in order to count the Incidents

Analysis (Hypothesis Q1.3)

In order to determine if there is a strong correlation between implementing changes and the number of Incidents the following approach is applied:

1. Count the number of Incidents made for a given product before changes are implemented for that given product
2. Count the number of Incidents made for a given product after changes are implemented for that given product
3. Compare the number of Incidents for a given product before and after implementing changes in order to see if there is a correlation between those actions.

Result (Hypothesis Q1.4)

After filtering the dataset to contain only the Incidents where the related product eventually will be subject for a Change, the number of Incidents is 11483 distributed on 177 products. Since the Incident events represent cases (1 event per Interaction) the count of Incidents is a feasible way of looking at the data from a cases perspective.

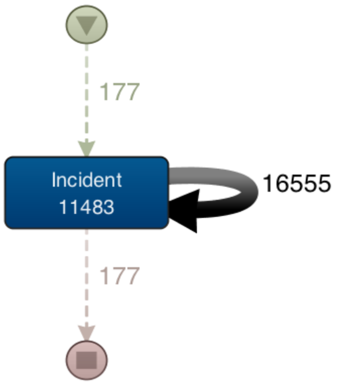


Figure 16 - Counting Interactions Before A Related Change Occur

Now, when looking at the number of Incidents after first change has occurred to the products (see Figure 18) they are related to, the number of Incidents increases from 11483 (see figure above) to 12241 (see Figure 18) which is an increase in the number of Incidents by 758 (6,6%).

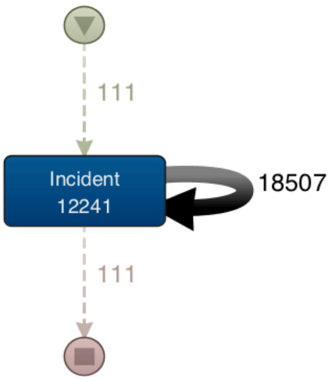


Figure 17 - Counting Interaction After First Occurrence of A Related Change

As it can also be seen (in Figure 18), the number of products having Incidents related to it goes down from 177 before implementing changes to 111 after implementing changes, meaning 66 (37,3 %) of the products do not have any related Incidents after changes has been implemented.

Seeing an increase in Incidents by 6,6% does not imply a strong correlation between implementing Changes and the number of Incidents.

Another perspective on the analysis regarding the correlation between implementing changes and the number of Incidents can be seen below.

Below is the total cases of the concatenated Changes and Incidents (Case view). Most cases with changes are not followed by a huge number of closed Incidents. This is due to most changes originating from Problem Management without many related Incidents.

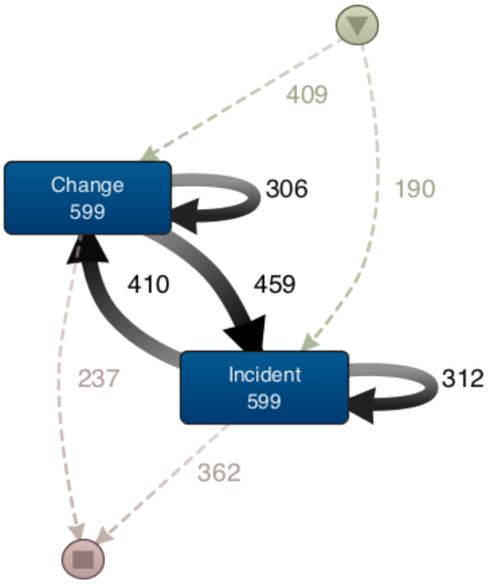


Figure 18 - Concatenated Changes And Incidents

Seen from an everything-else-equal perspective it is evident that the number of Interactions increases after implementing Changes. However, the increase in Incidents by 6,6% does not imply a strong correlation.

The missing strong descrease in Incidents after implementing Changes is not caused by bad quality in the implemented Changes, but it is proposed to examine if implementing Changes causes bugs elsewhere in the software product.

Subquestion 2 – Returning To A Steady State

The subquestion 2 “Returning To A Steady State” is divided into two detailed subquestions:

Subquestion 2 Divided Into Two Detailed subquestions

- Q2.1 - What is the average period to return to a steady state?
- Q2.2 - What is the average increase/decrease of Closed Interactions once a new steady state is reached?

In order to answer those two questions it is needed to determine what constitutes a “Steady state”. In the context of this paper, it is assumed that a steady state is determined by a period in time (of at least a few days) with approximately the same distribution of Incidents per day.

Looking at the absolute frequency of Changes and Incidents events (see below), it seems like Changes are not in general followed by a lot of closed Interactions but that the number of Interactions is still high.

Data To Be Analysed (Subquestion 2 – Q2.1)

The concatenated Changes and Incidents are imported into Disco with Type (Change or Incidents) as Activity, “CI Name (aff)” as ID, and the date fields mentioned in assumption 3 in the Assumptions Section as time stamps. This generated 1 case for each Product with Changes and Incidents as activities.

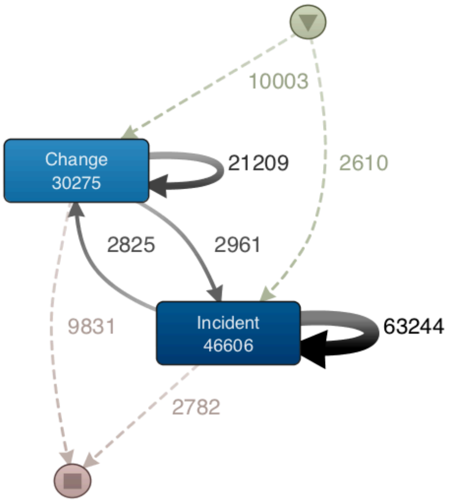


Figure 19 - Process Map For Product Cases Having Changes / Incidents

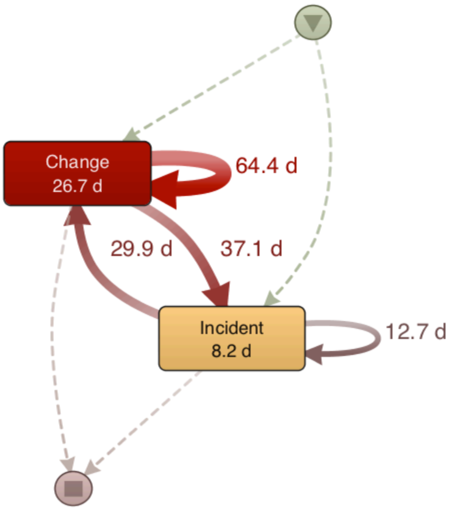


Figure 20 - Performance Map Mean Duration - Product Cases Having Changes / Incidents

By looking at the average time between events it can be seen that the average time between Incidents (12,7 days) is significant lower than the average time from Implementing a Change to the next Incident (37,1 days).

It seems like the number of Incidents are not significantly affected by Changes but that there is a decrease the first couple of days and then going back to a stable level (indicated by the average time between Incident→Incident is significantly higher than the average time between Change→Incident.

To determine the average increase/decrease of Closed Interactions once a new steady state is reached, it is needed to look at the average number before implementing Changes and the average number after implementing Changes. This can be determined by the average time between Incidents (Figure 20), but it can also be determined by counting the number of Incidents closed before and after Changes as done in subquestion 1. In subquestion 1 the assumption is that a collection of changes constitutes a Change (see Figure 3). However, another perspective can be taken by looking at the average number of Interactions closed on days where Changes occur, compared to the number of Interactions closed on days where no Changes occur.

It is assumed here that implementing Changes only affects the number of Interactions on the day the Change is implemented. By looking at those numbers it can be determined by how much the number of Interactions increases/decreases when Changes are implemented[2]

[2] Of course other assumptions regarding period in time a Change affects Interactions closed could be measured. The dataset could be manipulated to

Data To Be Analysed (subquestion 2 – Q2.2)

The concatenated Changes and Interactions are imported into Disco with Type (Change or Interaction) as Activity, and the date fields mentioned in assumption 3 in the Assumptions Section as time stamps. The ID is a concatenation of ID+CloseTime in order to generate 1 case for each Product on a Given day (with at least 1 event). This way it is possible to look at days with or without Changes for a given Product.

Data Filtering (subquestion 2 – Q2.2)

In order to find the Interactions on a day where changes has been implemented, the data (as described in the "Data To Be Analysed (subquestion2 – Q2.2)" are filtered so that:

- Remove cases not having any Change events. In order to get only days with Changes
- Remove cases not having any Interaction events. Cases consisting of only Changes are not interesting in this analysis
- Remove all Change activities in order to count the Interactions

In order to find the Interactions on a day where no changes has been implemented, the data (as described in the "Data To Be Analysed (subquestion 2 – Q2.2)" are filtered so that:

- Remove cases having any Change events. In order to get only days with Changes
- Remove cases not having any Interaction events. Cases consisting of only Changes are not interesting in this analysis
- Remove all Change activities in order to count the Interactions

show other aspects (e.g. that a Change relates to Interactions Closed within a week after a Change is implemented) and the analysis can then be re-run.

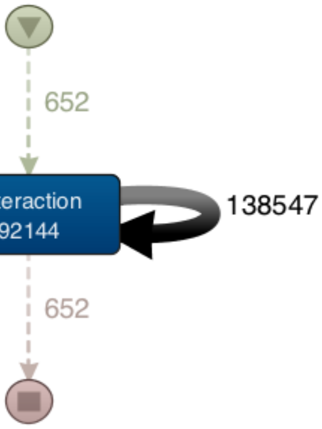


Figure 21 - Number Of Interactions Closed On Day Where Changes Have Occurred For The Product

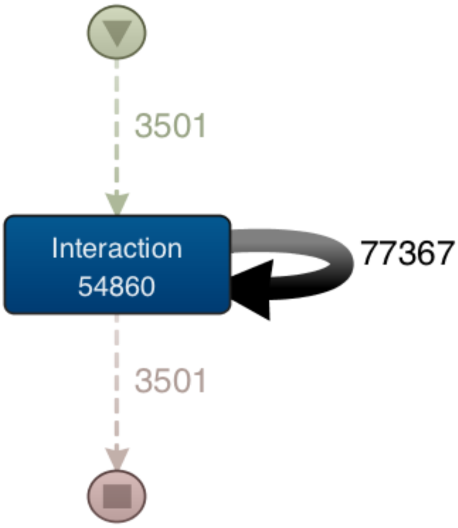


Figure 22 - Number Of Interactions Closed On A Normal Day Where No Changes Have Occurred For The Product

As it can be seen from Figure 21 a remarkably higher number of Interactions are closed on days where Changes are implemented compared to the number of Interactions closed on normal days where no Changes are implemented for the related product.

Subquestion 3 – Change in Average Steps to Resolution

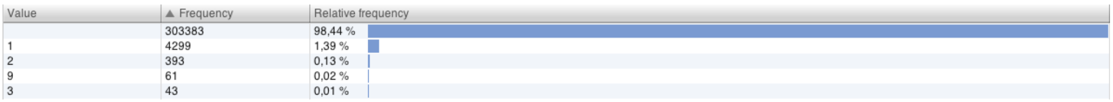
In this subquestion it is examined if the service delivered is the same after implementing Changes as it is before implementing Changes.

In order to do so, it is assumed that the following 3 performance parameters indicates if the service level is unchanged after Changes has been implemented:

1. **Number of Calls** - Looking at the number of calls (closed Interactions) before and after implementing Changes
2. **The Average number of steps it takes to solve Incidents** with related Changes compared to the average number of steps to solve Incidents with no related changes

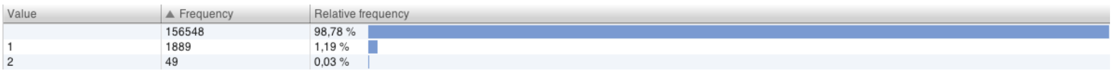
Ad Subquestion 3.1) By looking at the number of calls (Closed interactions) there is a remarkably increase in the number of closed Interactions after implementing Changes (see Figure 21) than before implementing Changes (Figure 22). This implies that the performance regarding handling calls (not getting new calls, but closing calls) is higher after implementing Changes.
NB: Data foundation and filtering equal to Subquestion 2 – Q2.2

Ad Subquestion 3.2) Looking at number of **steps** to solve Changes the "Detail Incident Activity" and "Detail Incident Merged" dataset has been analysed. It is assumed that filtering the dataset for Incidents related to some/none Changes (as specified in the dataset) gives a valid picture.



Value	▲ Frequency	Relative frequency
	303383	98,44 %
1	4299	1,39 %
2	393	0,13 %
9	61	0,02 %
3	43	0,01 %

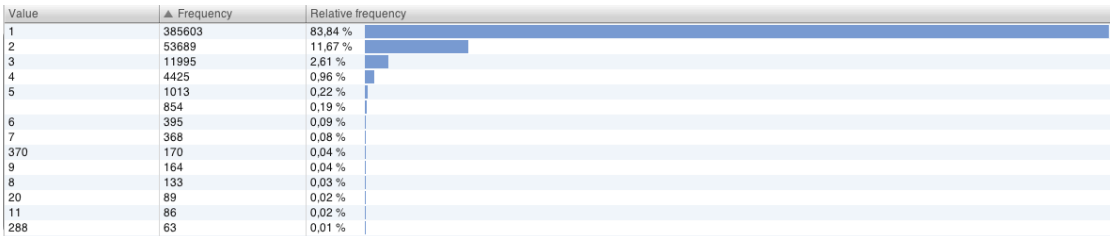
Figure 23 - Incidents With 9 Or More Activities - Distributed On Related Changes



Value	▲ Frequency	Relative frequency
	156548	98,78 %
1	1889	1,19 %
2	49	0,03 %

Figure 24 - Incidents With 8 Or Less Activities - Distributed On Related Changes

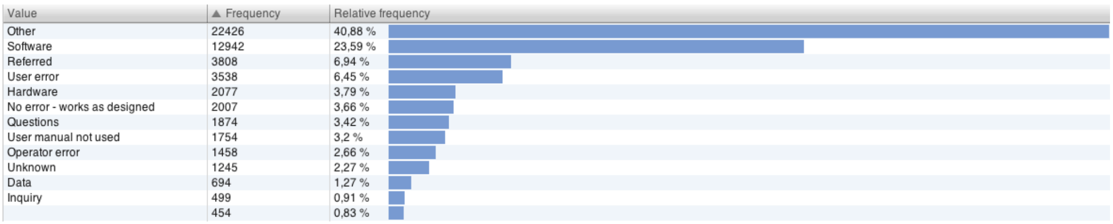
Based on the above two figures there is no evidence that Incidents related to Changes requires more steps to solve than solving Incidents with no Changes related. The number of steps required to solve an incident is more likely related to type of incident, complexity and related product than the related Changes.



Value	▲ Frequency	Relative frequency
1	385603	83,84 %
2	53689	11,67 %
3	11995	2,61 %
4	4425	0,96 %
5	1013	0,22 %
	854	0,19 %
6	395	0,09 %
7	368	0,08 %
370	170	0,04 %
9	164	0,04 %
8	133	0,03 %
20	89	0,02 %
11	86	0,02 %
288	63	0,01 %

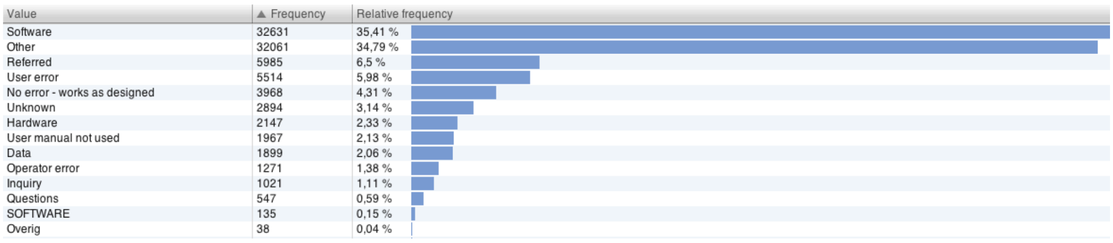
Figure 25 - Number Of Related Interactions For Incidents With No Changes Related

Getting into details on the Calls (Interactions closed) on a day with Changes implemented compared to a normal day with no Changes implemented:



Value	▲ Frequency	Relative frequency
Other	22426	40,88 %
Software	12942	23,59 %
Referred	3808	6,94 %
User error	3538	6,45 %
Hardware	2077	3,79 %
No error - works as designed	2007	3,66 %
Questions	1874	3,42 %
User manual not used	1754	3,2 %
Operator error	1458	2,66 %
Unknown	1245	2,27 %
Data	694	1,27 %
Inquiry	499	0,91 %
	454	0,83 %

Figure 26 - Closed Calls (Interactions) On A Normal Day With No Changes Implemented - Distributed On Closure Code



Value	▲ Frequency	Relative frequency
Software	32631	35,41 %
Other	32061	34,79 %
Referred	5985	6,5 %
User error	5514	5,98 %
No error - works as designed	3968	4,31 %
Unknown	2894	3,14 %
Hardware	2147	2,33 %
User manual not used	1967	2,13 %
Data	1899	2,06 %
Operator error	1271	1,38 %
Inquiry	1021	1,11 %
Questions	547	0,59 %
SOFTWARE	135	0,15 %
Overig	38	0,04 %

Figure 27 - Closed Calls (Interactions) On A Day With Changes Implemented - Distributed On Closure Code

It is interesting to see that software related Interactions increases significantly after implementing Changes, while hardware related Interactions is almost the same. This could indicate that the implemented Changes are implemented correct (solving the initial problem), but that implemented changes in software either 1) causes new issues or 2) Leads the attention to other issue that needs to be solved (issues that were present all the time). This needs to be investigated further by Rabobank.

Subquestion 4 – Creativity

In these section discoveries in the Change process ("Change Process – Process View" dataset) is analysed using Disco.

Data Filtering (subquestion 4 (part 1)

- Cases not having a Change Record Close Time stamp (indicates incomplete cases, since this is set automatically)
- Cases completed after 30.04.2014 (indicates incorrect data)

After filtering 97% of the cases remain for further analysis.

13% of all (the filtered) Changes have an Actual End (and often also Actual Start) later than the Requested End (requested by the customer. This could cause bad customer experience and recreation of new Incidents and new Interactions.

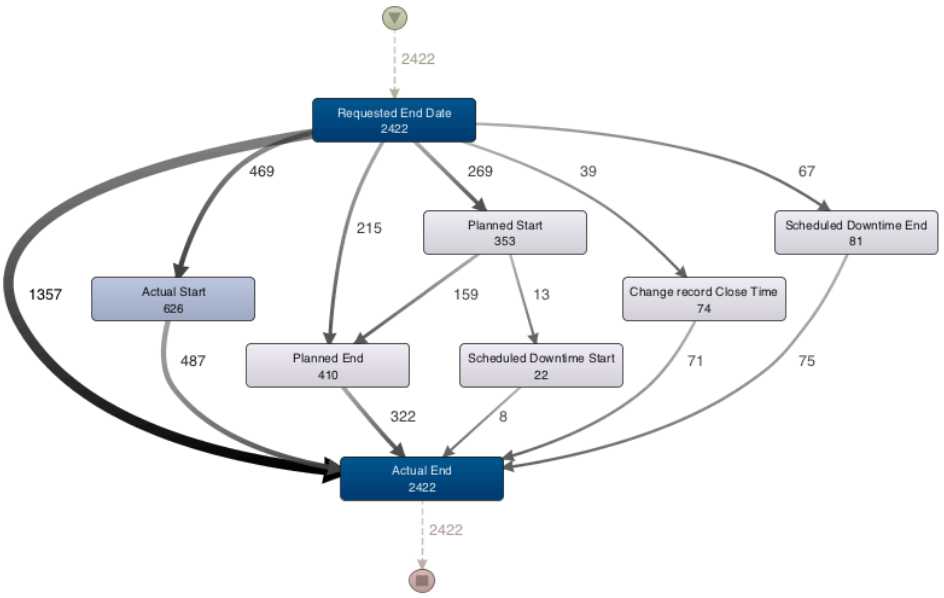
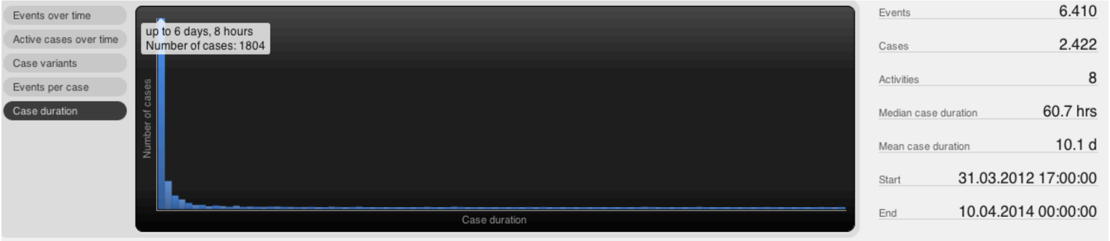


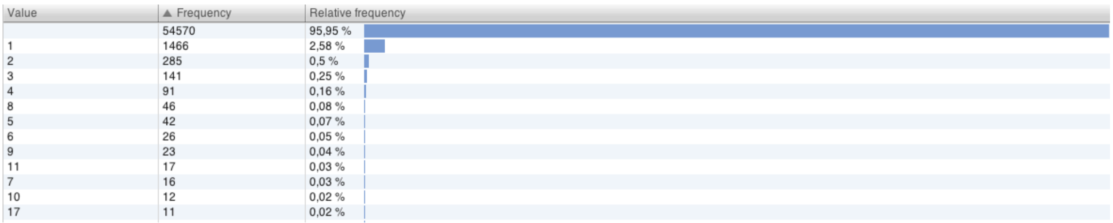
Figure 28 - Number Of Cases Where Actual End Is Later Than Requested End (Trimmed To Events ActualEnd -> RequestedEnd)

In average the Actual End is 10,1 days after the requested end as seen in the statistic below.



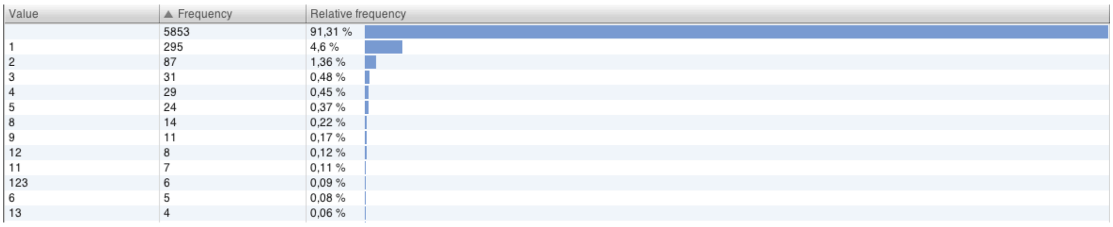
Events	6.410
Cases	2.422
Activities	8
Median case duration	60.7 hrs
Mean case duration	10.1 d
Start	31.03.2012 17:00:00
End	10.04.2014 00:00:00

Figure 29 - Duration From Requested End To Actual End For Cases Where Actual End Is Later Than Requested End



Value	▲ Frequency	Relative frequency
	54570	95,95 %
1	1466	2,58 %
2	285	0,5 %
3	141	0,25 %
4	91	0,16 %
8	46	0,08 %
5	42	0,07 %
6	26	0,05 %
9	23	0,04 %
11	17	0,03 %
7	16	0,03 %
10	12	0,02 %
17	11	0,02 %

Figure 30 - Number Of Related Incidents Related To Changes Solved Before Requested End



Value	▲ Frequency	Relative frequency
	5853	91,31 %
1	295	4,6 %
2	87	1,36 %
3	31	0,48 %
4	29	0,45 %
5	24	0,37 %
8	14	0,22 %
9	11	0,17 %
12	8	0,12 %
11	7	0,11 %
123	6	0,09 %
6	5	0,08 %
13	4	0,06 %

Figure 31 - Number Of Related Incidents Related To Changes Solved Later Then Requested End

As it can be seen from Figure 30 and Figure 31 the number of Changes being related to Incidents is relatively higher if the Change implementation is later than requested. Being late generates more Incidents.

Another perspective analysed is the relationship between Incidents closed and opened before implementing Changes, compared to Incidents closed and opened after implementing Changes.

The number of Incidents opened per Product per day where no Changes has been implemented for that Product is 42775/16098 (see Figure 33) = 2,7 Incident.

If that number is compared to the number of Incidents opened per Product per day where Changes are indeed implemented for that product 3632/860=4,2 Incident (see Figure 32).

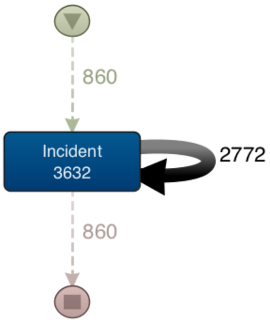


Figure 32 - Number of Incidents Opened On Days Where Changes Are Implemented

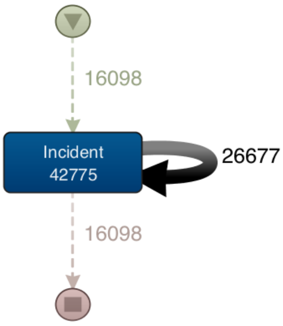


Figure 33 - Number Of Incidents Opened On A Normal Day With No Changes Implemented

It seems like the implementation of Changes initially solves the issues, but quickly the number of Incidents opened is back on a high level. How can that be? Is it because too few Incidents are closed or because relatively more Incidents are opened? This is looked into details below:

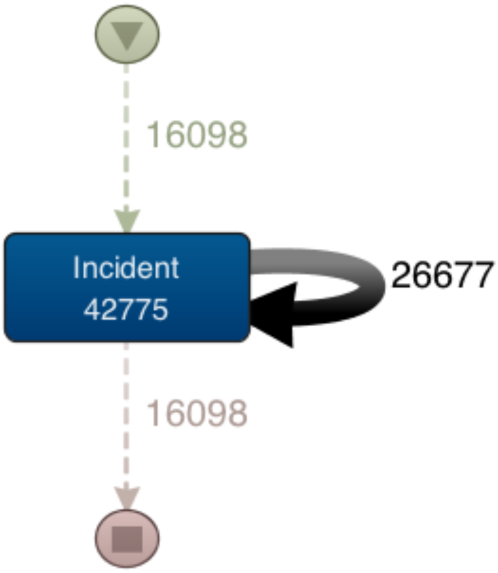


Figure 34 - Number Of Incidents Closed On Day Of Changes

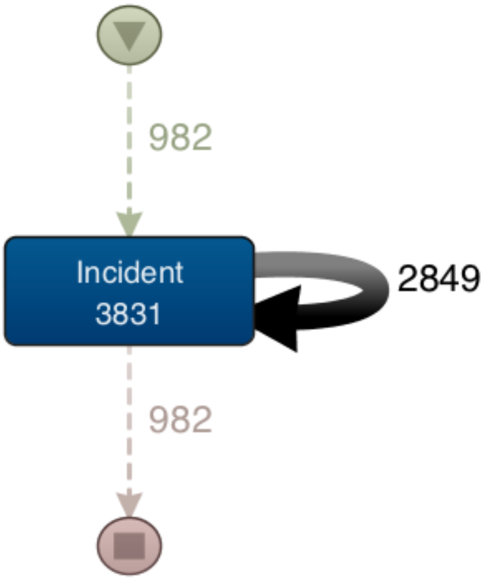


Figure 35 - Number Of Incidents Closed On Day With Related Changes

Number of Incidents closed per Product per day where no Changes has been implemented for that Product is 42775/16098 = 2,7 Incident (Figure 34).

That number is compared to the number of Incidents closed per Product per day where Changes are indeed implemented for that product (3831/982=3,9 Incident (Figure 35).

It is interesting that on average more Incidents are closed on days with Changes (3,9) than on a normal day without changes (2,7). However, on days with Changes in average 4,2 Incidents are opened (compared to only 2,7 Incidents being opened in average on normal days with no Changes). Closing 3,9 Incidents on average per product on days with related changes is good, but since 4,2 new changes are also opened it means that on days with Changes on average the number of Incidents have increased by 7,7%.

What is the reason for that? It could be that implementing Changes might solve initial reported issues but the implementation introduces new Incidents. It could also be that the Incidents directly related to the code changes are indeed closed, but there are continuously more and more other Incidents being identified in the products. A red flag is raised, but to conclude, Rabobank needs to investigate this in more details.

For example the level/details of requirements could be benchmarked to see if certain level of requirements leads to introducing more issues after implementation.