

Support och underhåll av Java-baserade serversystem

Kim Slotte

Datavetenskap

Naturvetenskaper och teknik, Åbo Akademi

Handledare: Dragos Truscan

Maj 2018

Referat

IT Service Management ramverk, processer och arbetssätt kan tillämpas på underhåll och support av Java-baserade serversystem. Ett av dom mest kända ramverken heter ITIL, som lämpar sig väl för att bygga ett koncept för ändamålet.

För ett fungerande koncept behövs grundläggande saker vara väl definierade, bl.a. underhållsavtal, roller, koordinering och arbetsflöden. Konceptet delas in i tre delar enligt källan varifrån ett ärende eller händelse initieras. Dessa tre delar är:

- ärendehantering, som initieras av kunden eller användaren till systemet
- övervakning, som initieras genom teknisk avlyssning av systemet
- stödande processer, som initieras av underhållspersonalen

Konceptet har processer och en tillhörande underhållsplattform. Underhållsplattformen består av applikationer som stöder underhållsarbetet. Följande processer ingår: ändringshantering, incidenthantering, problemantering, administrativa begär, användarsupport, övervakning, referentgranskning, konfigurationshantering och leveranshantering.

Sökord: ITIL, IT service management, Java server, underhåll, support, övervakning, helpdesk

Innehållsförteckning

1	Inledning	1
2	Koncept, ramverk och standarder	1
2.1	IT Service Management.....	1
2.2	ITIL	1
2.3	Andra ramverk, koncept och standarder	2
3	Allmänt gällande support och underhåll	2
3.1	Terminologi	4
4	Java-baserade serversystem.....	4
4.1	Systemmiljö och artefakter	4
5	Underhållsarbetet.....	5
5.1	Underhållsavtal	5
5.2	Roller	6
5.3	Koordinering.....	6
5.4	Arbetsflöde.....	8
6	Ärendehantering.....	8
6.1	Incidenthantering (process)	8
6.2	Ändringshantering (process)	9
6.3	Administrativa begär (process)	10
6.4	Användarsupport (process).....	10
7	Service desk	11
7.1	Servicedesk implementation.....	11
7.2	Skapande av ärenden	12
8	Övervakning.....	13
8.1	Teknisk övervakning (process)	13
8.2	Planera övervakning.....	14
8.3	Hitta övervakningsmål.....	15
8.4	Väsentliga övervakningsmål.....	16
8.5	Referentgranskning (process)	17
9	Problemantering (process)	18

10	Konfigurationshantering (process)	18
10.1	Konfigurationshanteringssystemet	18
10.2	Körtida inställningar och filer	19
11	Leveranshantering (process)	20
11.1	Kontinuerlig leverans.....	20
12	Avslutning	21
13	Källor	22

1 Inledning

Denna kandidatavhandling handlar om support och underhåll av Java-baserade serversystem och dess källkod. Genom källkoden kan affärssystem ändras, byggas och installeras inom en relativ kort varsel. Att serversystemet kan ha fel och vara ostabilt beaktas.

Olika service management-ramverk och principer kommer att användas som grund för att planera ett optimalt och effektivt sätt för support och underhåll. Det planerade optimala sättet kommer att utgöra ett servicekoncept som kan erbjudas som en IT-leverantör. Här kommer dock endast att anges den teoretiska designen för de väsentliga delarna ur service management, utan att desto mera gå in på konkret implementation med olika applikationer och verktyg. Lösningen kommer att vara en kombination av processer och arbetssätt. Kort kommer även att nämnas vilka system som behövs för att stöda processerna och arbetssätten.

Servicekonceptet kommer att tillämpas för Java-baserade serversystem. Dock med minimala förändringar kan konceptet tillämpas på annan teknologi, t.ex. .NET eller PHP. Konceptet strävar till att uppnå en nivå tre enligt Capability Maturity Model Integration (CMMI), där processerna är väl definierade.

2 Koncept, ramverk och standarder

2.1 IT Service Management

IT Service Management är ett allmänt koncept för olika processer och arbetssätt för att sköta om IT-tjänster. Meningen är att få arbetet mer strukturerat, optimerat, uppföljbart, verifierbart, spårbart och kontrollbart. Genom konceptet strävar man till att uppnå högra kvalitet och kanstandseffektivitet för de IT-tjänster som levereras. IT service management kan formas i olika ramverk, bestående av olika processer. (Haverblad, s. 18-20) Ett sådant ramverk är till t.ex. ITIL.

2.2 ITIL

Information Technology Infrastructure Library (förkortas ITIL) är ett processramverk för att organisera IT-arbetet, även kallad IT service management (förkortas ITSM). Under 1980-talet utvecklade brittiska myndigheter en samling av rekommendationer och praxis för sin IT-hantering. Denna samling har sedan dess utvecklats och fått namnet ITIL. ITIL fungerar väl som ett gränssnitt och kontrollmekanism mellan leverantör och köpare. (Görling, s. 227-229) Under år 2007 utgavs

version 3 av ITIL, bestående av fem bokvolymerna om totalt ca. 1500 sidor. (ITIL Official Introduction, 2007)

2.3 Andra ramverk, koncept och standarder

Det finns andra ramverk och standarder som kan tillämpas för underhåll av Java-baserade serversystem. Följande är kända:

- ISO/IEC 20000 är en internationell standard för IT service management. Företag kan certifiera sig enligt denna standard. Att få fram gemensam terminologi har varit ett delmål med standarden. (Kunas, s. 13)
- Kontinuerlig leverans (eng. Continuous Delivery) strävar att kunna bygga, testa, och leverera programvara på kort varsel. (Swartout, s. 14)
- DevOps strävar att få programvaruutvecklare och systemadministratörer att jobba i harmoni utan eller med låga organisatoriska hinder. (Swartout, s. 14)

Vid ytterligare optimering, utvidgning och förbättring av processerna och arbetssätten kan även följande vara intressanta:

- Lean IT handlar om att optimera procesströmmen inom IT service management. Optimering sker genom att ta bort s.k. spill (eng. waste). (Edmead)
- Six Sigma, strävar att förbättra kvaliteten genom processers output. Standarden strävar till att minimera fel och minimera processvariansen. (Kunas, s. 34)
- CMMI (Capability Maturity Model Integration), strävar att förbättra processkvaliteten och därmed även förbättra service kvaliteten. (Kunas, s.35)

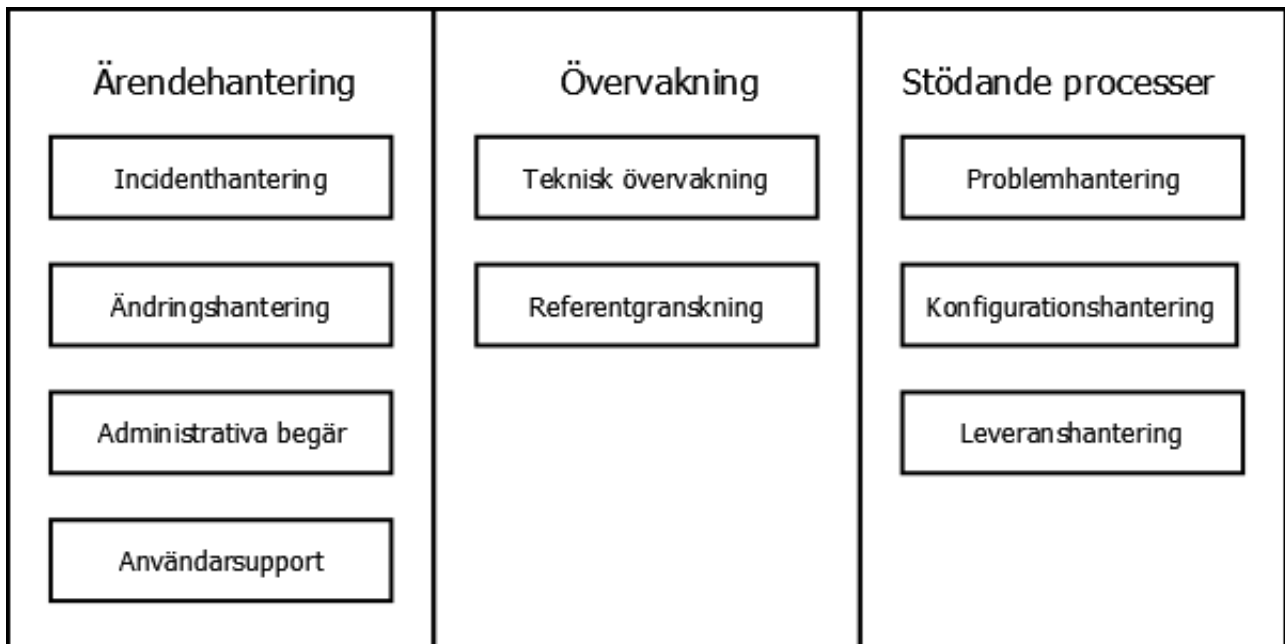
3 Allmänt gällande support och underhåll

Som ett centralt begrepp för de typer av uppgifter, som utförs inom support och underhåll, kallas i denna text för ärenden. Dessa ärenden hanteras olika beroende på process och typ. I denna implementation av support och underhåll finns följande processer:

- Ändringshantering
- Incidenthantering
- Administrativa begär
- Användarsupport
- Teknisk övervakning

- Referentgranskning
- Problemhantering
- Konfigurationshantering
- Leveranshantering

I följande figur har processerna delats in i grupperingarna ärendehantering, övervakning och stödande processer:



Som stöd för processerna finns en underhållsplattform som innehåller information, applikationer och verktyg. I denna implementation har följande huvudsystem:

- ärendehanteringssystem, där olika ärenden kan registreras och hanteras
- innehållshanteringssystem, för att tillhandahålla information för underhållsarbetet
- versionshanteringssystem, för att tillhandahålla källkod och konfigurationer av teknisk karaktär
- underhållsserver, som innehåller mindre komponenter för integrations och automatisering av processer
- övervakningssystem, för att automatisera och rapportera potentiella fel
- service desk, tjänst där kunden och slutanvändare initierar sina ärenden
- kontinuerligt integrationssystem, för att bygga system från källkod och installera byggda paket

Delar in de mänskliga resurserna i tre grupper:

1. *Underhållspersonalen* sköter om support och underhållet för systemet.
2. *Kunden* äger, erbjuder och ansvarar för systemet.
3. *Användare* som använder systemet.

3.1 Terminologi

Support och underhåll har en del special terminologi. För att enklare förstå sammanhang och betydelse finns en lista ("ITIL ordlista och förkortningar") på webbadressen:

https://www.axelos.com/Corporate/media/Files/Glossaries/ITIL_2011_Glossary_SV-v1-0.pdf

4 Java-baserade serversystem

Java-baserade serversystem är i huvudsak programvara som körs i en applikationsserver baserad på Java Enterprise teknologi. Java Enterprise är en uppsättning komponenter och standarder för nätverksapplikation och tjänster. Det finns olika applikationsserver leverantörer som uppfyller standarden. De mest vanliga applikationsservrar är dock Apache Tomcat, Oracle Application Server, WebSphere och JBoss.

Arkitekturen för Java-baserade serversystem är uppbyggt enligt följande (de tidigare är beroende av de senare i listan): applikationer, applikationsplattformen, applikationsservern, Java virtuella maskinen, operativsystemet, server och nätverket.

Servicekonceptet och processerna i detta dokument kommer att begränsa sig till applikationerna, applikationsplattformen, applikationsservern och Java virtuella maskinen. Operativsystemet, server och nätverk kommer att lämnas utanför servicekonceptet. Gällande applikationslivscykeln kommer fokus att ligga på underhållet. Underhållsfasen har uppnåtts när systemet inte mera kontinuerligt utvecklas eller utvecklingstempot har avsevärt minskat.

4.1 Systemmiljö och artefakter

För ett Java-baserat serversystem har vi följande delar och artefakter:

- Källkod, som beskriver systemets logik i programmeringsspråket Java
- Installationspaket, är ett paket av kompilerad källkod som kan installeras
- Server, där installationspaket kan installeras för att bli exekverade
- Konfigurationsfiler, som tillhör servern och möjliggör att systemet fungerar på önskat sätt

- Databas, där data sparas för serversystemen
- Data, system tillhandahåller data som bör sparas i en databas eller i filsystemet
- Dokumentation, systemets tekniska och funktionella dokumentation

Oftast uppehålls skilda servrar för utveckling, testning och produktions användning.

5 Underhållsarbetet

För att ha ett välfungerande underhåll för systemen bör några aspekter vara klargjorda. Nedan beskrivs underhållsavtal, roller och koordinering.

5.1 Underhållsavtal

Ett underhållsavtal ingås mellan kund och IT-leverantör, antingen som ett formellt avtal eller som överenskommet arbets sätt. När ett avtal för underhåll (Haverblad, s. 363-370 / Kunas, s. 64-65) görs bör det ingå följande:

- lista över de serversystem delar som täcks av underhållet
- servicenivå för serversystemen
- underhållspersonalens tillgänglighet och tjänstetider
- ansvar hos leverantören
- pris för underhållet
- regler, tidsramar och befogenheter (se detaljer nedan)

Regler, tidsramar och befogenheter för ärenden kan eventuellt variera enligt prioritet och vem som initierat ärendet. Dessa definieras efter kundens önskemål. (Haverblad, s. 365 / Kunas, s. 64-65) Reglerna har dock följande karaktär:

- Om ett ärende får påbörjas utan godkännande eller krävs ett godkännande från kunden?
- Hur mycket tid får användas för ett påbörjat ärende, fören explicit godkännande från kunden krävs? Här har erfarenhet visat att tre timmar fungerar väl som den minimala tidsramen. I de flesta fall har underhållsleverantören fått en uppfattning om omfattningen gällande ärendet inom den tidsramen. De mer triviala ärendena kan även genomföras inom den tidsramen.
- Ett tidsestimat ges om ett ärende överskrider det som krävs för ett explicit godkännande. Om ett tidsestimat överskrider mycket krävs möjligen nytt godkännande från kunden. En procentuell tidsrams gräns på t.ex. 30 procent kan definieras.

- Får ett ärende påbörjas utan godkännande eller kräver det godkännande från kunden.
- Den totala tidsanvändning för ett system som får användas.
- Kräver driftstopp för leveranser ett explicit godkännande? I så fall enligt vilka situationer, så som leverans tidpunkt, driftsstoppets längd, leveransens risknivå och det underhållna systemets status?

Ovanstående regler kan vara definierade olika beroende på de system som underhålls.

5.2 Roller

IT-leverantörens underhållschef fungerar som gränssnitt till kunden. Från kundens sida utdelas ansvarsrollen applikationsägare, så att alla applikationer och komponenter har någon som är ansvarig för underhållet. Underhållschefen kommunicerar främst med applikationsägaren för olika typ av koordinering, godkännanden och rapportering av ärenden. Underhållschefens huvuduppgift är att koordinera arbetet (se Koordinering) och vara den drivande kraften för ärendet. Själva tekniska arbetet sköts av IT-leverantörens underhållspersonal eller av en tredje part. Eskalering till en tredje part används i situationer då det tekniska arbetet ligger utanför de tekniska befogenheter som getts leverantören. Eskalering kan även behövas vid behov av specialkunskap.

Ovanstående rollstruktur har valts för att underhållschefen kan enklare få en helhetsbild över situationen i systemen vid kritiska lägen. Underhållspersonalen kan fokusera på att lösa specifika tekniska (del-)fall utan att behöva förstå helheten. Rollstrukturen är baserad på erfarenhet från arbete med Java-baserade serversystem.

5.3 Koordinering

De flesta ärenden i detta dokument innebär någon typ av koordinering främst på underhållschefens ansvar. Till koordineringen (ITIL Service Transition, s. 49-56 / ITIL Service Operation, s. 49-, 110 / ISO/IEC 20000-1:2000, s. 21-22) innebär att

- identifiera vilka tekniska komponenter (både interna och externa) som påverkas av ärendet
- identifiera vilken miljö och system version som ärendet gäller
- evaluera ärendets omfattning, risk, kostnader, tidsestimat och tidtabell
- allokerar resurser för att få ärendet avklarat
- göra upp plan på hur ärendet skall lösas
- informera hur ärendet fortskrider

- prioritera ärendet och se till att tidsramar med svarstid och lösningstid följs (se nedan)
- se till att den slutliga ändringen dokumenteras (se Ändringshantering)
- verifiera och testa lösningen på ärendet
- leverera en eventuell ändring till systemet (se Leveranshantering)
- stängning av ärendet när de involverade har slutfört sina uppgifter

För prioritering av ärenden tillämpas IEEE Standard 1044.1-1995 (IEEE Standard 1044.1-1995, s.35) och ISO/IEC 20000-2:2012 (ISO/IEC 20000-2, s. 60) med passande beskrivning enligt Java-baserade serversystem:

- **Kritisk:** Systemets väsentliga delar fungerar inte alls. Arbetet gällande ärendet börjar genast och fortsätter tills en lösning hittats.
- **Hög:** Systemets användbarhet är väsentligt begränsat. Arbetet gällande ärendet är viktigt och en lösning kan förväntas inom några dagar.
- **Medium:** Systemets användning begränsas måttligt. En lösning på ärendet kan förväntas inom några veckor.
- **Låg:** Systemet har ett fel, som inte orsakar begränsningar i användbarheten. En lösning på ärendet kan förväntas inom några månader.

Svarstid under kontorstid väljs till en timme. I normala kontorsomgivningar är en sådan svarstid rimlig. Lösningstiden finns beskrivna ovan i samband med prioriteringar. Lösningstiden varierar pga. av den stora variationen av ärenden och möjligheten att ha en jämn arbetsbörda för underhållspersonalen. Något ärende kan omfatta en liten ändring, medan ett annat ärende kan omfatta en större uppgradering av hela systemet.

Ärenden dokumenteras (ITIL Service Operation, s. 53) både tekniskt och funktionellt på följande sätt:

- beskrivning av incidenten, ändringen, begäran eller problemet
- plan hur ärendet kommer att lösas
- beskrivning hur ärendet har blivit genomfört

5.4 Arbetsflöde

För arbetsflödet behövs en process som följs. Olika typer av ärenden kan ha olika typer av processer. (ITIL Service Transition, s. 46-48) Följande bör ingå:

- steg och delmoment i kronologisk ordning (arbetsflöde) att genomföra ärendet
- ansvariga som genomför ärendet
- tidsramar och tidströsklar som de ansvarig tillämpar
- eskalationsregler för vem som kontaktas i olika special situationer

Vi tillämpar ett enkelt arbetsflöde med följande delmoment:

- Öppen: ärendet har registrerats men hantering av ärendet har inte påbörjats
- Väntar: ärendet väntar på att något förrän arbetet fortsätter
- Under arbete: hantering av ärendet pågår
- Genomfört: en lösning på ärendet har getts
- Stängt: ärendet stängs efter lösningen har blivit accepterad

6 Ärendehantering

Vi har ett flertal processer som har likande typ av hantering av ärenden. Denna gruppering kallas här ärendehantering för att binda dom likartade samman. Gemensamt för dessa att de oftast registreras av en person i servicedesken.

Vid registrering av ärenden är följande information väsentlig (ITIL Service Operation, s. 49, 54):

- unikt nummer för ärendet
- tid och datum när ärendet registrerats
- kontaktperson som registrerat ärendet
- prioritet enligt kontaktpersonen
- beskrivning av ärendet

6.1 Incidenthantering (process)

Incidenthantering innebär att hantera de eventuella fel som uppstår i systemet under dess användning. Målet är att återställa systemet till normal operations nivå så snabbt som möjligt. En incident initieras av systemets användare eller via en kontakt person. (ITIL Service Operation s. 46-47)

För incidenter bör beskrivningar innehålla följande vid registrering:

- tidpunkt då incidenten uppstod
- information om i vilket system incidenten uppstod
- beskrivning av händelseförlopp när incidenten uppstod
- eventuella felmeddelanden

Underhållspersonalen undersöker incidenten och tar fram en lösning för att återställa operationsnivån i systemet. Undersökningen och lösningen är av teknisk karaktär, där personalen undersöker dokumentation, källkoden, system konfigurationer, system data och servermiljön (se Systemmiljö och artefakter). Ärendet överförs på problemhantering, om incident har ett underliggande större problem. Om problemet är trivialt hanteras ärendet inom incidenthantering.

6.2 Ändringshantering (process)

Ändringshantering arbetsprocessen ger metoder och procedur för att hantera effektivt alla ändringar i ett system. Ändringarna registreras i ett ändringshanteringssystem av applikationsägare och underhållspersonal. Målet är att ändringarna registreras, evalueras, auktoriseras, prioriteras, planeras, testas, implementeras och dokumenteras. En ändring kan vara proaktiv eller reaktiv. (ITIL Service Transition, s. 42-44, 53). För serversystem betyder proaktiva ändringar en ny egenskap; medan reaktiva ändringar betyder ett fel i systemet (även kallad bugg). En reaktiv ändring kan även uppstå i situationer då systemets omgivning ändras; i sådana fall där det finns ett beroende mellan olika delar av systemet.

Service desken (se Service desk) fungerar som gränssnitt för applikationsägare och andra kundkontakter för att initiera en ändring. Ett ändringsprojekt skapas om en ändring är för omfattande att genomföra som ett enskilt ärende. Ändringsprojekten har olika processer och leveransartefakter beroende på typ. Följande ändringprojekt har identifierats:

- integrering, för att koppla samman system
- utveckling, för att vidare utveckla systemet mer omfattande
- migration, för att flytta, ändra eller uppgradera systemplattformen
- granskning, kartlägga och granska systemet för något ändamål

6.3 Administrativa begär (process)

Kunden och systemets slutanvändare kan ha en del små begär som inte är incidenter eller ändringar. (ITIL Service Operation, s. 55-59) Vid behov görs webbformulär för hantering av begär som behövs hanteras separat, t.ex. behörighetshantering. Andra administrativa begär kan vara ändringar i serversystemets data, där genomförandet av ändringen kräver administrativa rättigheter.

Behörighetshantering handlar om att ge en användare behörighet till ett system. Detta innebär att ge auktoriserade användare behörighet medan icke-auktoriserade användare förhindras behörighet. (ITIL Service Operation, s. 68)

För behörighetshantering skräddarsys enligt kund och system. (ITIL Service Operation, s. 69)

Följande bör beaktas och finnas dokumenterat i innehållshanteringssystemet:

- tillvägagångssättet och process för att begära behörighet
- godkännande av en behörighetsförfrågan
- uppehålla en struktur över roller, grupper och användarspecifika kopplingar för systemen

6.4 Användarsupport (process)

Inkluderar som en viktig process i konceptet, att ge systemets slutanvändare support och hjälp i situation de behöver. Vikten av att sträcka konceptet till slutanvändare är att förstå vilka användarsvårigheter det finns med systemet. Med hjälp av att kartlägga dessa svårigheter kan problem identifieras genom referentgranskningen (se Referentgranskning). Problemen hanteras då via problemhanteringen.

Till teamet som sköter användarsupporten rekommenderas både personal från kunden och IT-leverantören. Dessa två personal gruppering har olika synvinklar. Informations utbyte och förståelse ger en bättre helhet. Kunden har en mer affärsinriktad synvinkel, medan IT-leverantören har en mer teknisk synvinkel. Verktyg för att sköta supporten erbjuds genom service desken (se nästa avsnitt) av IT-leverantören.

För användarsupporten används samma arbetssätt som för andra ärenden. Skillnaden är dock, att här initieras ärendet av användaren istället för kunden. Hantering av ärendet delegeras direkt till underhållspersonalen av underhållschefen. Underhållschefens roll är mer passiv vid användarsupport.

7 Service desk






Service deskens idé är att erbjuda en primär kontaktpunkt för att rapportera och initiera ärenden. Kontaktpunkten kan bestå av telefonsamtal, e-post, web formulär eller automatisera händelser. Denna idé om en primär kontaktpunkt kalls på engelska för "Single Point of Contact". En service desk kan formas på många olika sätt. (ITIL Service Operation, s. 108-117) En service desk kan innehålla information och verktyg för att underlätta supporten av ärenden, t.ex. självhjälp funktioner, hjälpinformation, kända fel och tillhörande lösningar, fjärrkontrollstyrning och diagnostikprogram. (ITIL Service Operation, s. 157)




7.1 Servicedesk implementation

För Java-baserade serversystem väljs som kontaktpunkt en web portal, där de initierade ärendena registreras i ärendehanteringssystemet. Vid kritiska situationer kan även kontakt tas per telefon. E-post kan även användas. E-post integreras med service desk portalen och ärendehanteringssystemet. Eftersom ärenden sällan er triviala, rekommenderas det att ärenden registreras skriftligt via e-post eller portalen. Service desken fungerar som ett gränssnitt att få registrerat ärenden till ärendehanteringssystemet, där underhållspersonalen behandlar ärenden. Service desken har främst två mål grupper; kundkontakter och systemets slutanvändare. Dessa två målgrupper har olika behov gällande support. För att enklare kunna uppfylla behoven för båda målgrupperna, görs två olika web portaler.

e-tuki Technical Support

[Lue tämä sivu suomeksi...](#)

Send support request	
<p>You can contact us with the support request form, by e-mail or by telephone. The issue can be about incident, enhancement or other issue.</p>	
	<p>Support request form In addition to a description you can fill the priority and type of the support request to the form. Files can also be easily attached to the request.</p>
	<p>Support to e-mail address request at e-tuki.fi A support request is automatically generated and forwarded to the correct responsible person from a message sent by e-mail.</p>
	<p>Support by phone You can call us if your issue is urgent or requires further clarification.</p> <ul style="list-style-type: none"> • Kim Slotte (Service Manager), +358 50 762 2404 • Matti Jormakka (Software Specialist), +358 40 7707 243
<p>Support request is a request for support sent by the customer which covers customer service and tracking. Issues and possible projects are founded for the actual actions.</p>	
Uutiset	
<p> e-tuki Support Center opens (03.02.2009)</p> <p> e-tuki Remote Control at your service (03.02.2009)</p>	

Applications	
	<p>Issue Center Through our Issue Center you see what issues and projects we are working on. You are able to see progress, work hours done and resolutions. Open application...</p>
	<p>Meeting Center Meeting Center provides a virtual conferencing environment on the Internet that enables interaction via speech, instant messaging and video. It is possible to show and manage documents and programs with other participants through it. Read more...</p>
	<p>Remote Control A small remote desktop application that allows us to connect to your computer's desktop. The application can either be installed or run as such. Read more...</p> <ul style="list-style-type: none"> • Download and run • Download and install

e-tuki Technical Support	
<p>e-tuki Technical Support is a portal service for supporting data systems. It is managed by Slotte Consulting, their customers and their cooperation partners to provide overall solutions for management and administration of server side data systems. Portal's purpose is to give a single access point for all parties involved in the support process; administrators, technicians, managers and other support personnel.</p>	

Ett exempel på en engelskspråkig web portal, som fungerar som service desk. (källa www.e-tuki.fi)

7.2 Skapande av ärenden

Enligt ITIL kan man ha olika typer av ändringar (ITIL Service Transition, s. 46). En ändring som registreras av kunden använder annan terminologi än underhållsleverantören. Därför skapas två olika typer av ändringar; ändringsbegäran för kunden och ändringsärende för underhållspersonalen. Kunden kan begära något som leder till flera (eller inga) ändringsärenden. Underhållsleverantören skriver då en funktionell-teknisk beskrivning i form av ändringsärende baserad på kundens begäran. Dessa länkas ihop i ändringshanteringssystemet.

Service deskens uppgift är att stå kunden till tjänst och informera om situationen (mao. kundbetjäning). Däremot hanteringen av ändringsärenden i ärendehanteringssystem har en mer funktionell-teknisk information som inte är väsentlig för kunden. Kunden kan i vissa fall inte formulera sina ändringar fullständigt. Därför bör en funktionell-teknisk tolkning göras från ändringsbegär till ändringsärenden. Notera även att ett flertal ändringsbegär kan vara likartade och därför bilda ett gemensamt ändringsärende.

8 Övervakning

Ett sätt att upptäcka problem i serversystem är att använda ett övervakningssystem eller göra manuella granskningar. Teknisk information om status och tillgänglighet hämtas genom övervakningssystemet. (ITIL Service Operation, s. 36-37) För Java-baserade serversystem har följande väsentliga situationer identifierats:

- ett system behöver vara i ett specifikt tillstånd eller mellan vissa gränsvärden
- en automatisk ändring är gjord i systemets
- användare använder systemet som utlöser en händelse

8.1 Teknisk övervakning (process)

Ett system är konfigurerat att ge ut information i form av händelser. Händelserna kan antingen analyseras automatiskt eller manuellt. I automatisk analys följs regler som har blivit definierade. Huvudsakliga automationen är filtrering enligt hur signifikant en händelse är. Automationen kan utvidgas till någon typ av korrelationsmotor, som kan avgöra liknande mönster och trender. (ITIL Service Operation, s. 39-40) ITIL föreslår tre nivåer av signifikans:

- Information, en händelse som endast informerar att något specifikt hände
- Varning, en händelse som varnar om något fel, som inte orsakar avbrott
- Avvikelse, en händelse där ett avbrott har skett

En trigger avlöses, om korrelations aktiviteter identifierar något som bör undersökas. Triggande handlar om att göra en händelse synbar för den personal som underhåller systemet. Det kan t.ex. vara ett larm, ett meddelande i ett övervakningssystem eller ett e-postmeddelande. Personalen bör analysera och möjligen lösa problemet som uppstår från dessa triggers. (ITIL Service Operation, s. 41-42)

Sammanfattning över hur en händelse hanteras kronologiskt:

- Upptäcka: händelser upptäcks i systemet
- Filtrering: händelser filtreras baserat på signifikans
- Korrelation: en korrelationsmotor hittar mönster och trender
- Triggande: en trigger avlöses, om åtgärder kan behövas
- Hantering: analysering och lösning av problemen av personal

8.2 Planera övervakning

En av de viktigaste aspekterna för att planera övervakning är instrumentering. Enligt ITIL handlar instrumentering om hur ett system blir övervakat. (ITIL Service Operation, s. 45) Följande huvud aspekter bör beaktas vid planering:

- gränssnitt som övervakas
- typ av övervakning
- regler för händelseprocessen
- informationen inkluderade i händelserna
- personalen som analyserar händelserna

Dessutom bör planering ske hur händelserna genereras och hanteras. System kan redan ha gränssnitt som ger ut händelser. I andra fall bör kanske någon typ av polling ske. Skräddarsydda komponenter och övervakningsagenter kan installeras för få fram önskade händelser. En övervakningsagent är en lätt klient programvarukomponent som kommunicerar med övervakningssystemet. (ITIL Service Operation, s. 45) Även följande bör beaktas:

- olika händelse typer har olika parametrar och information
- händelse data bör ha en plats för lagring
- händelser bör möjligtvis dirigeras (eng. ruoting) och transporteras genom flera steg

Huvuddelarna att planera generering och hantering handlar om följande:

1. identifiera gränssnitt
2. möjligen lägg till skräddarsydda komponenter eller övervakningsagenter
3. samla och hantera händelserna som genereras

Det finns två huvudtyper för övervakning; aktiv och passiv övervakning. Aktiv övervakning interagerar eller använder test data med systemet. Passiv övervakning lyssnar däremot endast på systemet. En indelning på reaktiv och proaktiv görs också inom övervakning. Med reaktiv övervakning menas att händelsen innehåller i sig all nödvändig information. Med proaktiv övervakning menas däremot att dra slutsatser från händelsemönster och trender. (ITIL Service Operation, s. 88)

Ett optimalt övervakningssystem samlar ihop alla händelser till en enda applikation.

Övervakningen blir enklare om händelserna finns samlade, istället för att vara utspridda över flera applikationer. Samla ihop alla händelser är dock en stor utmaning vid planeringen av övervakning.

8.3 Hitta övervakningsmål

Hitta vad som skall övervakas är väsentligt. Det kräver planering för att undvika falska uppfattningar om situationen hos systemen. Målet är att hitta så många problem som möjligt i ett tidigt skede. (Lehmann-Haupt) Olika gränssnitt, händelser och mätetal bör hittas. Alexis Lê-Quôc på Datadog definierar mätetal enligt; mätetal fångar ett värde tillhörande ett system vid en specifik tidpunkt. Mätetal har en specifik frekvens för att samlas från systemen. (Lê-Quôc)

Enligt Noah Lehmann-Haupt på Scalyr kan övervakningsmål delas enligt:

- Potentiellt dåligt händelser, som borde triggas före något dåligt händer. Dessa är resurser som kan ta slut, bli fyllda eller överflöda. Triggande bör ske före en kritisk nivå har uppnåtts.
- Aktuellt dåliga händelser, som borde triggas då något dåligt händer. Händelsen bör meddelas underhålls personalen snarast möjliga.
- Goda händelser, som borde triggas då dåliga händelser slutar hända. Denna information visar att ett system är operativt och används.

Ett system har flera arkitekturlager. Varje lager bör övervakas separat för att kunna identifiera vilket lager som har en möjlig avvikelse eller är i ett icke-operativt läge. (Lehmann-Haupt) Lagren är enligt följande:

- applikation, där logiken exekveras
- process, som håller applikationen körbar
- server, en container för processer och applikationer
- hosting leverantör, erbjuder servern och håller den kopplad till internet
- externa beroenden, som t.ex. SSL certifikat och domän namn
- användaren, som använder applikationen

8.4 Väsentliga övervakningsmål

Baserad på tidigare teori och tips från olika källor har följande övervakningsmål identifierats för Java-baserade serversystem:

Målkategori	Mätvärde(n)	Arkitekturlager
Java virtual machine	Java heap användning, Java garbage collection overhead, aktiva trådar (Subramanian)	Applikation
Applikations integritet	Verifiera att källkoden är intakt	Applikation
Applikations integritet	Simulera kontakt med andra integrerade system och komponenter.	Applikation
Applikations integritet	Ändringar av körtida filer	Applikation
Licenser	Verifiera äktheten och giltigheten hos applikationen och andra komponenter	Applikation
Fel, varningar och information	Applikations logfiler	Applikation
Applikations tillgänglighet	Process tillgänglighet (Subramanian)	Process
Server resurser	CPU, disk utrymme, minne, nätverks belastning, I/O operationer (Lehmann-Haupt)	Server
Applikations tillgänglighet	Response time (Subramanian)	Hosting leverantör
Domän namn	Domän namns sökning	Externa beroenden
SSL certifikat	Verifiera äkthet och giltighet	Externa beroenden
Applikations användning	Antal inlägg registrerade eller hanterade i applikationen	Användaren
Applikations tillgänglighet	Regressions testning av att öppna applikationen för att verifiera auktorisering	Användaren

Tabell fortsätter på nästa sida.

Målkategori	Mätvärde(n)	Arkitekturlager
Applikations tillgänglighet	Regressions testning av kopplingar till integrerade system och komponenter	Användaren
Applikations användning	Tillgänglighets loggar (eng. access log files)	Användaren
Applikations säkerhet	Applikations intrång	Användaren

8.5 Referentgranskning (process)

Övervakningsprocessen behöver förbättring genom att justera och omdefiniera regler. Något väsentligt kan missas av den tekniska övervakningen om reglerna inte är rätt inställda. Pga. detta bör en granskning ske som här kallar referentgranskning. (ITIL Service Operation, s. 39-42 / Lehmann-Haupt) I denna granskning bör tidigare händelser granskas och analyseras. En granskning kan även ske helt skilt från teknisk övervakning, genom att manuellt granska systemet tekniskt. Manuell granskning kan behövas för situationer som inte täcks av övervakningssystemet.

Med hjälp av "IEEE Standard for Software Reviews and Audits" (IEEE Std 1028™-2008) planeras granskningens struktur. Idén är att granska all källor av all anknutna händelser till det underhållna systemet. För förverkliga detta behövs lista över källor och en granskningsfrekvens. Även att processen granskar sig själv borde vara inkluderat. Varje källa bör ha beskrivning hur granskning skall ske. Ett tidsintervall för granskningens mål bör även anges. Det är inte nödvändigtvis ändamålsenligt att granska alla händelser.

Resultatet från granskningen är en rapport över hittade avvikelser, gjorda åtgärder och förslag på åtgärder. Under granskning bör följande tas i beaktan:

- Hitta en lämplig nivå för konfigurationer av händelse filter, korrelation och triggande. En lämplig nivå är när endast väsentliga händelser når underhållspersonalen.
- Hitta avvikelser som indikerar fel i systemet. För varje fel bör en enkel analys över signifikans och allvarlighetsgrad göras. Enkla fel kan åtgärdas i samband med analysen. Ett förslag på åtgärder görs om mer omfattande fel. Dessa fel registreras som problem till problemhanteringen.

9 Problemhantering (process)

Problemhantering strävar till att hitta de underliggande felen i ett system. Dessa fel kallas problem. Problemhanteringsärenden skapas främst via incidenthanteringen (se Incidenthantering) och övervakning (se Övervakning). Målet är förebygga att liknande händelser sker i framtiden (Kunas, s. 89-90). Det tekniska arbetssättet för underhållspersonalen är likt incidenthantering.

Notera att proaktiv problemhantering täcks i denna processtruktur av referentgranskningsprocessen (se Referentgranskning), jämfört med en del rekommendationer inom service management.

10 Konfigurationshantering (process)

Konfigurationshantering innebär att uppehålla aktuell information om systemets komponenter och deras relationer. Informationen bör även ha en ändringshistoria. Förutom arkitektur och integrations relationer bör även ansvariga roller för komponenten anges. (ITIL Service Transition, s. 65-68) Informationsnivån bör åtminstone vara på en sådan nivå att underhållspersonal kan få en överblick av hur ett system är uppbyggt och fungerar. Underhållschefen ser till att nivån hålls på en rimlig nivå. Nivån är en balans mellan informationstillgänglighet, kontroll och insats för uppehållande av informationen (ITIL Service Transition, s. 73). Varje komponent bör ha en väl definierad identifikation och ändringshistoria. Ändringshistoria kan även vara anknuten till version information. (ITIL Service Transition, s. 74-75)

Aktuell information kan (tillämpas enligt situation) för Java-baserade serversystem t.ex. vara:

- versionsnummer för installerat system
- filkatalog där systemet är installerat
- webbadressen för tjänsten som systemet erbjuder
- adress till tekniskt gränssnitt till autentisering
- konfigurationsfil för systemet

10.1 Konfigurationshanteringssystemet

I samband med en reaktiv ändring i ett system (se Ändringshantering) bör konfigurationshanteringssystemet uppdateras. Konfigurationshanteringssystemet roll är att uppehålla den tekniska informationen om det underhållna systemet. (ITIL Service Transition, s. 68)

För automatiserad proaktiv konfigurationshantering se nästa avsnitt.

Som primärt system för hantering av konfigurationer har ett innehållshanteringssystem valts. I ett innehållshanteringssystem kan ändringar göras dynamiskt och historieinformation bibehålls. Till systemet kan dock ha referenser till annan typ av information som till exempel filer, internet adresser, källkod och komponent relationer. Eftersom innehållshanteringssystemet är det primära systemet för konfigurations hantering bör varje komponent vara registrerat och dokumenterat i systemet. I vissa fall kan dokumentationen vara kortfattad.

Listor på ändringshistoria kan skrivas in i innehållshanteringssystemet som en ändringslog (tidpunkt och vad som gjorts) eller som referens till ändringshanteringssystemet. Konfigurationer i formen av kod och skript sparas i däremot i ett versionshanteringssystem. Även referenser och beskrivningar till gällande kod och skript bör finnas i innehållshanteringssystemet.

10.2 Körtida inställningar och filer

Körtida inställningar och filer borde även täckas av konfigurationshantering. Lösningen är att lägga in väsentliga filer i versionshanteringssystemet. Även en databasdump borde göras om väsentlig information finns lagrad i databasen. Den automatiserade tekniska lösning här består av versionshanteringssystemet, som initieras med ett cronjob-tidskörning från underhållsservern. Med lämpliga intervall matas inställningar och filer från det underhållna serversystemet till versionshanteringssystemet.

Denna uppläggnig ger möjlighet att kolla upp ändringshistoria enligt aspekterna vad som ändrats och när något ändrats (med noggrannheten av valt intervall). Vi har dock ingen koppling till vem som gjort ändringen i underhållna serversystemet. Men eftersom underhållspersonalen brukar vara få till antalet borde det gå att utreda på annat sätt vid behov, t.ex. genom att fråga de involverade. Även ur datasäkerhetssynvinkel kan integriteten och oavvislighet övervakas. Om ett spionprogram installeras kommer det att med stor sannolikhet upptäckas. Detta upplägg ger oss även möjlighet att verifiera att den noll tolerans (enligt ITIL) för att kringgå ärendehantering verkligen följs (ITIL Service Transition, s. 45).

Installation och driften av den tekniska lösningen hör till arbetsprocessen ändringshantering. Medan analysen och mindre åtgärder av de aviserade meddelandena hör till arbetsprocessen problemhantering (se Problemhantering). Underhållschefen ansvarar för att väsentlig information täcks av ändringshanteringen.

11 Leveranshantering (process)

En leverans görs i då en ändring kräver en mer omfattande ombyggnad av systemet, som t.ex. en ny installation. Inom leveranshantering är målet a bygga, testa och leverera en ändring till ett system. Följande delar (ITIL Service Transition, s. 84-86) är väsentlig för Java-baserade serversystem:

- tids-planera leveransen
- bygga installationspaket för leveransen
- testning av installationspaket
- verifiera kompatibilitet gällande beroenden till andra komponenter
- riskanalys gällande installationen
- uppgöra en ånger plan om installationen misslyckas
- instruktioner för installationsprocessen
- validering av installation

För koordineringen av ovanstående ansvarar underhållschefen, som i sin tur håller kontakt med applikationsägaren hos kunden. Till kundkoordineringen hör främst följande; vad som levereras, leveranstidtabellen och eventuellt planerat fönster för avbruten drift. För att göra leveranser systematiskt görs en leveransrapport där ovanstående delar finns klargjorda.

11.1 Kontinuerlig leverans

För att kunna leverera på kort varsel används konceptet kontinuerlig leverans (eng. Continuous Delivery, förkortad CD). För att uppnå konceptet (Swartout, s.71-72) bör följande eftersträvas för underhåll av Java-baserade serversystem:

- användning av versionshanteringssystem
- ändra små kodsegment ofta
- gör inte koden för komplex
- bygg systemet från källkod ofta
- kör automatiserade testfall ofta

För att automatisera ovanstående används ett kontinuerligt integrationssystem (eng. Continuous Integration, förkortad CI). (Swartout, s. 77) Notera, att övervaknings processen (se Väsentliga övervakningsmål) inkluderar övervakning som bygger och testar systemet ("Verifiera att källkoden är intakt") via det kontinuerliga integrationssystemet.

12 Avslutning

Olika IT service management koncept, ramverk och arbetssätt har tillämpats för Java-baserade serversystem. De mest väsentliga delarna för ett fungerade support och underhåll har täckts mycket väl.

Ur kritisk synvinkel kan det vara svårt att förstå vikten av de olika processerna och deras proportioner. Kortfattad beskrivning av processen betyder inte att processen har minimal betydelse. Möjligen kan helheten vara svår att förstå av någon, som inte är bekant med IT-arbete. Dock har vikten lagts mer på att få ett heltäckande koncept, än att beskriva terminologi och motiveringar.

Konceptet kunde utvidgas genom att även ta med vissa delar ur andra mindre viktiga processer, t.ex. kapacitetshantering. Även underhållsplattformen kunde beskrivas i mer detalj och beskriva en mer konkret väg att implementera konceptet. Att bygga upp och installera underhållsplattformen kunde dessutom ge klara indikationer vilka delar av processerna som är möjligen ofullständiga. En annan dimension av utvidgning är att gå en nivå djupare på CMMI-skalan (Capability Maturity Model Integration), till nivå fyra (förvaltd) eller fem (optimerad).

13 Källor

www.e-tuki.fi, Marknadsföringsmaterial från Oy Slotte Consulting Ab:s tjänst www.e-tuki.fi (2009)

Mark Edmead, daterat 21.7.2015 (läst 3.4.2018), *Lean IT and DevOps: The new kids on the block* <<https://www.cio.com/article/2900051/agile-development/lean-it-and-devops-the-new-kids-on-the-block.html>>

IEEE Guide to Classification for Software Anomalies (IEEE Std 1044.1-1995), Institute of Electrical and Electronics Engineers Inc., 1996

IEEE Standard—Adoption of ISO/IEC 20000-1:2011, Information technology—Service management— Part 1: Service management system requirements, IEEE Computer Society, 2013

IEEE Standard—Adoption of ISO/IEC 20000-2:2012, Information technology—Service management— Part 2: Guidance on the application of service management systems, IEEE

IEEE Standard for Software Reviews and Audits (IEEE Std 1028™-2008), Institute of Electrical and Electronics Engineers Inc., 2008

ITIL Official Introduction to the ITIL Service Cycle, The Stationery Office, 2007

ITIL Service Operation, The Stationery Office, 2007

ITIL Service Transition, The Stationery Office, 2007

Stefan Görling, *Att arbeta med IT Projekt*, Studentlitteratur, 2009

Angelica Haverblad, *IT Service Management i praktiken*, Studentlitteratur, 2007

Michael Kunas, *Implementing Service Quality based on ISO/IEC 20000 : A Management Guide*, IT Governance Publishing, 2012

Noah Lehmann-Haupt, daterat 10.3.2017 (läst 24.6.2016), *Scalyr: Zen and the Art of System Monitoring* <<https://www.scalyr.com/community/guides/zen-and-the-art-of-system-monitoring>>

Alexis Lê-Quôc, daterat 30.6.2015 (läst 24.6.2016) *Datadog: Monitoring 101: Collecting the right data* <<https://www.datadoghq.com/blog/monitoring-101-collecting-data/>>

Karun Subramanian, daterat 29.5.2014 (läst 24.6.2016), *5 Metrics you MUST monitor in your Java Application* < <http://karunsubramanian.com/jee/5-metrics-you-must-monitor-in-your-java-application/>>

Karun Subramanian, läst 24.6.2016, *Is log analysis an effective approach to application performance management?* <<http://karunsubramanian.com/apm/is-log-analysis-an-effective-approach-to-applicationperformance-management/>>

Paul Swartout, *Continuous Delivery and DevOps : A Quickstart guide*, Packt Publishing Ltd., 2012
Computer Society, 2013