



Slutrapport NYS-projektet 2015-2018

Sammanställd av: Johan Andersson von Geijer, Birgitta Andréasson, Stefan Berndtsson,
Rebecka Gustafsson, Magnus Kull, Emma Ternrud, Sture Ågren

Augusti 2018

Innehåll

Inledning och bakgrund.....	2
Mål och effektmål	2
Projektets fyra faser	2
Projektets första tre faser	3
Fas 1 - förstudie	3
Fas 2 - vägval och utvärdering av funktioner	4
Fas 3 - upphandling	5
Fas 4 – utveckling och implementation.....	5
Processarbete.....	6
Implementation av Primo.....	6
Koha och utveckling.....	7
Servermiljö och migrering	10
Förberedelser för implementation.....	11
Detaljplanering av systembytet.....	12
Förvaltning av Koha och Primo.....	13
Vad har vi lärt oss?	13
Vad gjorde vi bra?.....	14
Vad kunde vi gjort bättre?.....	15
Överraskningar?	15
Bilaga 1: Projektbemanning	17
Bilaga 2: Systemskiss	18
Bilaga 3: Teknisk dokumentation	19

Inledning och bakgrund

I januari 2015 påbörjade en arbetsgrupp förberedelserna för ett större systembyte på Göteborgs universitetsbibliotek (GUB). Arbetet drevs i projektform, under namnet NYS-projektet (Nya system), och projektuppdraget var att ersätta äldre systemlösningar. De system som ingick var biblioteksdatasystemet Virtua, discoverysystemet Summon, länkserver SFX samt ERM-systemet Verde. Systemen hade otaliga kopplingar med varandra och med andra system.

Följande problem med de befintliga systemen kunde identifieras:

- Ingen vidareutveckling skedde
- Bristande möjligheter till anpassningar eller förändringar
- Stora mängder data flyttades mellan system
- Systemen stödde inte dagens processer och materialtyper

Den samlade effekten var att informationssökning blev en onödigt besvärlig process för bibliotekets slutanvändare. Nya tekniska lösningar fanns som gav bättre användarupplevelser. Det fanns också krav på konkurrensutsättning från upphandlingssektionen för flera av de system som ingick i projektet.

Mål och effektmål

Målet var att ersätta befintliga discovery- och biblioteksdatasystem, samt länkserver.

Följande effektmål definierades inledningsvis:

- En systemlösning som ger en förbättrad användarupplevelse för bibliotekets slutanvändare
- Upplevd förbättrad datakvalitet
- En systemlösning som innebär möjlighet till tydliga och effektiva arbetsprocesser för bibliotekets medarbetare
- Framtidssäkra möjligheten till automatiserad integration med kringliggande system, och användande av ny teknik
- Underlätta framtida upphandlingsprocesser

Projektets fyra faser

Projektgruppen bestod av personer med olika kompetenser och under projektets senare del utökades arbetsgruppen med ytterligare kompetenser. Projektgruppen rapporterade regelbundet till en styrgrupp. För detaljer om bemanning av projektet se bilaga 1.

NYS-projektet genomfördes i fyra faser under åren 2015-2018.

Fas 1 – förstudie

Projektgruppen tog fram olika handlingsalternativ och analyserade dem utifrån bibliotekets behov. Gruppen beslöt att systemet Verde skulle utgå ur projektet.

Fas 2 – vägval

På projektgruppens rekommendation fattade styrgruppen ett inriktningsbeslut om att välja open-

sourcelösningen Koha för att ersätta befintligt ILS. En utvärdering av funktionalitet i Koha genomfördes för att bedöma om systemet skulle uppfylla GUB:s behov. Vidare utarbetades en kravspecifikation som underlag för upphandling av discoverysystem och länkserver.

Fas 3 – upphandling

Discoverysystem och länkserver upphandlades och resultatet blev systemen Primo och SFX, båda från Ex Libris.

Fas 4 – implementation och driftsättning

Inledningsvis gjordes en omfattande processkartläggning, därefter implementerades och driftsattes Primo och Koha efter utveckling och anpassning av och kring dessa system.

Under hela projektperioden har GU:s rekommenderade projektmodell PPS använts.

Projektets första tre faser

Fas 1 - förstudie

Projektets fas 1 pågick under första halvåret 2015 och hade som uppdrag att ge ett väl underbyggt underlag till beslut om vägval av systemlösning.

Projektgruppen bestod under fas 1 av sju personer med olika funktioner inom GUB. Under denna fas lades mycket tid på olika former av omvärldsbevakning i form av möten och diskussioner med andra universitets- och högskolebibliotek som också befann sig i olika stadier av översyn av sitt systemlandskap. Gruppen genomförde en enkät om funktionen i nuvarande system bland GUB:s medarbetare och hade också en workshop om olika discoverysystem. Resultatet från enkäten och workshopen användes dels i förstudien men blev också viktigt att foga in i arbetet med upphandling i fas 2 och 3. Under fas 1 tittade gruppen också närmare på ett antal system genom provperioder. Detta gav gruppen möjlighet att bilda sig en bättre uppfattning om hur systemen fungerade och hur olika arbetsflöden var konstruerade. I omvärldsbevakningen ingick också litteraturstudier.

Projektgruppen kom under denna fas fram till följande tre alternativ:

Alternativ 1: ILS och discovery/länkserver i två upphandlingar efter varandra

Alternativ 2: ILS och discovery/länkserver i en upphandling

Alternativ 3: Open source som ILS och upphandling av discovery/länkserver

Projektgruppen förordade alternativ 3 då det kombinerade open source-systemets höga grad av anpassningsbarhet och kontroll över ett verksamhetskritiskt system med de förbättringar i datakvalitet och arbetsprocesser som discovery/länkserver från *en* leverantör förväntades ge. Projektgruppen var också tydlig med att bibliotekets användare i framtiden borde mötas av *ett* sammanhållet gränssnitt till bibliotekets resurser.

Projektgruppen ville också anpassa valet utifrån hänsynstaganden till integration mellan system, snabb rörlighet i systemval, ny teknik och tätare konkurrensutsättning i framtiden.

Projektgruppen rekommenderade att inte upphandla ett ERM-system. Anledningen var att GUB vid tidpunkten hade ett fungerande system med gällande avtal, samt att en upphandling skulle medföra

onödig komplexitet till nackdel för viktigare systemdelar. Arbetet med att ersätta GUB:s ERM fortsätter genom att undersöka lösningar som inte kräver upphandling, exempelvis open source eller icke kommersiella alternativ.

För en mer detaljerad redovisning av arbetet i fas 1 hänvisas till delrapporten *NYS 2017 fas 1: underlag för val av nya bibliotekssystem på GUB*.¹

Fas 2 - vägval och utvärdering av funktioner

Projektets fas 2 pågick under andra halvan av 2015 till februari 2016. Huvudsyftet under fas 2 var att utreda om Koha som system motsvarade GUB:s krav och behov.

Projektgruppen behövde undersöka vilken funktionalitet som fanns i Koha, testa att den tekniska kringutrustningen kunde samverka med systemet och inte minst klargöra hur våra arbetsflöden skulle komma att påverkas.

Ett antal workshops med kollegor hölls för att identifiera vilka behov och funktioner ett ILS måste uppfylla och vilket systemstöd olika delar av verksamheten behövde. Utifrån resultaten utarbetades arbetsflöden och testprotokoll som sedan användes för att få svar på ett stort antal frågor om systemet. En testinstallation av Koha sattes upp, med data migrerad från det befintliga ILS Virtua. Tester utifrån testprotokollen genomfördes på denna instans tillsammans med kollegor.

Resultatet av testerna blev att basfunktionalitet inom våra viktigaste områden ansågs fungera och att systemet i stora drag uppfyllde de behov som identifierats med hjälp av kollegor. Testerna visade också att viss funktionalitet saknades och behövde åtgärdas för att GUB skulle kunna använda Koha. Projektgruppen listade därför de viktigaste utvecklingsbehoven men också frågor som dykt upp som inte rörde utveckling utan mer verksamhetsförändringar som GUB behövde ta ställning till.

För en mer detaljerad redovisning av arbetet i fas 2 hänvisas till delrapporten *NYS 2017 fas 2: underlag för val av Koha som ILS på GUB*.²

Förutom tester av själva funktionaliteten i Koha fördes två viktiga diskussioner angående att välja ett open source-system. Den första handlade om vilka möjligheter som finns till både samarbete med och stöd från andra bibliotek. Den andra om hur vi skulle organisera arbetet kring ett system som är så verksamhetskritiskt.

Det finns en väl organiserad community kring Koha, med ett stort antal användare både i Sverige och internationellt.³ I Sverige använder både universitets- och högskolebibliotek, stora folkbibliotek och specialbibliotek Koha och bidrar till dess utveckling genom att införliva ny funktionalitet i systemet.⁴ Det finns också ett antal konsultföretag som arbetar riktat mot Koha. Koha arbetar med en sexmånaders release-cykel.

Projektgruppen förde också ingående diskussioner om hur GUB på bästa sätt ska organisera arbetet kring Koha. För frågor som rör exempelvis förvaltning, utveckling och systeminteraktion måste det finnas ett självklart forum för diskussion och beslut, när ett open source-system har implementerats. I systemförvaltningen måste det finnas möjlighet att hantera frågor och önskemål i flera steg: först föra en diskussion om huruvida ett förslag eller önskemål ska genomföras, vilka konsekvenser det får

¹ <http://koha.se/wiki/Dokumentation>

² <http://koha.se/wiki/Dokumentation>

³ <https://koha-community.org/>

⁴ <http://koha.se/>

för övriga system och sedan hur arbetet i sådana fall ska föras vidare och genomföras. Det måste finnas tydliga mandat att fatta beslut i frågor som rör systemet. Även frågan om dokumentation diskuterades och det konstaterades att det är mycket viktigt att det material som produceras av GUB samlas och tillhandahålls på ett konsekvent och öppet sätt.

Arbetet med kravspecifikation för discoveryssystem och länkserver påbörjades under hösten 2015 och upphandlingen annonserades i februari 2016.

Fas 3 - upphandling

Fas 3 av NYS-projektet utgjordes av upphandling av discoveryssystem och länkserver och pågick under tidsperioden februari till april 2016.

Efter upphandlingsbeslut i maj 2016 tecknades avtal i september 2016. Resultatet av upphandlingen innebar att GUB fick Ex Librissystemen Primo och SFX som SaaS (Software as a Service).

Fas 4 – utveckling och implementation

I rapporten för fas 2 rekommenderade projektgruppen att ta in en extern projektledare för arbetet i fas 4, vilket också skedde. Under fas 4 implementerades de två systemen Primo och Koha. Arbetet med Koha inleddes med en omfattande processkartläggning. I samband med detta arbete utökades projektgruppen med tre processägare. Implementationen av Primo och arbetet med utveckling av Koha påbörjades och drevs delvis parallellt.

Projektgruppen ansåg att NYS-projektet i det här läget behövde en projektledning med en projektledare som inte hade koppling till verksamheten, som kunde ägna sig uteslutande åt projektet och ta det övergripande projektledaransvaret. Uppdraget var att samordna de olika delprojekten, samt leda och genomföra arbetet med de interna processförändringarna.

Uppdraget bestod av de fyra delprojekten:

- Verksamhetsprojekt – processutveckling
- Implementation av länkserver och discoveryssystem
- Utveckling av Koha
- Implementation av Koha

Projektledaren påbörjade arbetet tidig vår 2016, men kom av icke arbetsrelaterade skäl att bytas ut under hösten samma år.

Den planerade tidpunkten för driftsättning av Primo och Koha kom efter hand att justeras vid ett par tillfällen. Den främsta anledningen till detta var att tidsramarna redan från början var snävt satta, men också svårigheten med rekrytering av nödvändig personal.

Primo implementerades och driftsattes februari 2017. Systembytet från Virtua till Koha gjordes under påskhelgen 2018 efter en omfattande förberedelse- och testperiod med utbildning av personal, flertalet testmigreringar, systemkonfiguration och utvecklingsarbete.

Processarbete

Processägare

I början av april 2016 utsågs processägare för de tre verksamhetsprocesserna förvärv, katalog och cirkulation. De skulle under och efter projektet ytterst ansvara för att processerna utvecklades, utvärderades och förbättrades. Processägarna skulle också fungera som expertstöd i det fortsatte arbetet med NYS-projektet och den kommande processkartläggningen, men också till systemägare och utvecklare, EÖG⁵ samt GUB:s ledning.

Processkartläggningen

I inledningen av fas 4 i NYS-projektet påbörjades en kartläggning av processerna förvärv, katalog och cirkulation, med syfte att beskriva, tydliggöra och efterhand effektivisera arbetsprocesserna. Processerna som kartlades utgick ifrån arbetet i det befintliga biblioteksdatasystemet Virtua. Processkartläggningen skulle fungera som ett underlag till de eventuella förändringar som en ny systemlösning kunde innebära. Processkartläggningen genomfördes under perioden april till september 2016.

Arbetet gjordes under ett antal arbetsmöten ledda av projektledaren där processägaren tillsammans med en arbetsgrupp gjorde djuplodande analyser av varje delmoment i de olika processerna. Därefter ritades och modellerades processerna och i ett senare skede gjordes även mer detaljerade beskrivningar.

Den processkartläggning som genomfördes under fas 4 kom senare att användas av projektets testgrupper i arbetet med att verifiera Kohas funktionalitet i de olika processerna.

Implementation av Primo

Implementationen av Primo genomfördes i ett samarbete mellan Ex Libris och GUB med start i oktober 2016. Avstämningsmöten mellan Ex Libris och GUB hölls varannan vecka via Skype.

Projektgruppen fyllde i detaljerade så kallade workbooks som Ex Libris sedan använde som underlag för att sätta upp GUB:s instans med rätt parametrar.

I slutet av oktober levererades testdata från Virtua och GUP⁶ till Ex Libris och GUB fick därefter tillgång till en så kallad sandbox där tester av dataöverföringen kunde genomföras.

Strax före jul 2016 var GUB:s Primoinstans laddad med kompletta produktionsdata från Virtua, GUP, GUPEA⁷ och SFX i den blivande produktionsmiljön.

Under hösten pågick arbete med att införa autentisering med CAS (central authentication service), i GUB:s fall x-konton och guskonton. Arbetet stötte på oväntade hinder och dessa problem löstes inte helt förrän strax innan driftsättning.

⁵ Enhetsövergripande grupper

⁶ Göteborgs Universitets Publikationer

⁷ Göteborgs Universitets Publikationer – Elektroniskt Arkiv

I början av januari ordnades Primoutbildning på plats i Göteborg för medarbetare som skulle arbeta med administration av systemet. Under utbildningen gjordes också ytterligare konfiguration av systemet.

GUB gick i produktion med Primo i början av februari 2017. Bytet av system från Summon till Primo var oproblematiskt. Databasen Kvinnsam lades till som datakälla i Primo direkt efter driftsättning.

Ett av GUB:s krav på det nya discoveriesystemet var att man skulle kunna se lånestatus för det tryckta materialet från bibliotekskatalogen direkt i Primo (real-time availability, RTA). Vid övergången till Koha planerades discoveriesystemet att bli användarnas enda gränssnitt mot det tryckta materialet och denna funktionalitet var därför mycket viktig. När GUB gick i produktion med Primo var RTA inte implementerat, det dröjde till mitten av mars innan det började fungera. Nästan omedelbart stängde GUB dock av medföljande RTA-funktionalitet då den var helt otillräcklig för GUB:s behov, bitvis direkt missvisande. Istället utvecklade GUB en egen RTA-funktion som kom på plats mycket snabbt. Denna funktionalitet var betydligt mer utvecklad än den av Ex Libris erbjudna, och i ett senare skede har den utvecklats vidare till exempel genom integrering med en ny beställningsapplikation.

Primo hämtade data och information om exemplarstatus från Virtua fram till dess att Koha togs i produktion. Efter bytet till Koha i början av april 2018 byttes datakällan Virtua och alla dess inställningar ut till motsvarande för Koha. Även exemplarstatus hämtas nu från Koha.

Under 2017 ordnades ett flertal utbildnings- och frågetillfällen för personalen, något som fortsatt under 2018.

Arbete med anpassningar och ändringar har pågått löpande sedan Primo implementerades. Efter bytet till Koha är en ny insats för att optimera Primo och anpassa framför allt sökavgränsningar planerad.

Koha och utveckling

GUB använder Koha för funktionerna cirkulation, katalogisering, periodikahantering och förvärv.

Kohas funktionalitet uppfyller väl de grundläggande funktionerna för GUB:s lånehantering, men flera av bibliotekets specialfall täcks inte av Kohas standardfunktionalitet utan har lösts i Koha på ett alternativt sätt. Ett exempel på detta är GUB:s så kallade "långlån" vilket i Koha har återskapats med hjälp av automatiska omlån.

Arbetet med utveckling i och kring Koha pågick från januari 2017 till januari 2018. Det utvecklingsarbete som gjordes gäller bland annat sökmotorn Elasticsearch, Librisimport, Edifact-beställningar, formulär för lånekortsregistrering, specialutformade framplökningslappar, gränssnitts Anpassningar samt integration med omgivande system som discovery, fjärrlånesystem, utskriftssystem, grupprumsbokning, offentliga datorer med mera.

Redan i en tidig fas av NYS-projektet påbörjades insamlandet av önskemål om utveckling och förändringar av Koha. För att hantera den stora mängden ärenden fick projektgruppen värdera och prioritera utifrån verksamhetsnytta. Så småningom resulterade detta i en lista, där de nödvändiga åtgärderna för att gå i drift var tydligt identifierade.

Utvecklingsarbetet på GUB sker iterativt enligt SCRUM-metoden. Efter varje avslutad sprint kunde förändringar i Koha och utvecklade applikationer testas av testgrupper som leddes av processägarna. Därefter återkopplades till produktägare och IT-utvecklare och på så sätt kunde dialogen mellan alla

berörda fortgå kontinuerligt. Denna dialog var avgörande för kvaliteten på slutresultatet. NYS-projektet har under hela perioden haft slutanvändaren och de webbaserade gränssnitten i fokus. Utveckling och anpassning har på så sätt gått hand i hand.

Egenutvecklade tillägg till Koha

Integration med discoveryssystem

GUB:s discoveryssystem Primo har integrerats med Koha via en egenutvecklad beställningsapplikation där låntagarna gör sina beställningar efter sökning. Dessutom hämtas exemplarlistor med aktuell tillgänglighetsinformation från Koha till Primo via ett anpassat CGI.

Beställningsapplikation

En större utvecklingsinsats har lagts på en separat beställningsapplikation för att låta användarna beställa fram material och köa på utlånat material. Detta grundar sig i GUB:s hantering av beställningar vilket inkluderar ett omfattande regelverk. GUB vill till exempel kunna begränsa beställningar av tillgängligt material baserat på lokalisering i stängt eller öppet magasin, vilket det inte finns inbyggt stöd för i Koha.

Det är viktigt att användarna möts av ett enhetligt gränssnitt vid reservation av tryckta böcker. Vid en reservation ska användaren inte behöva lämna discoverysystemet för att lägga reservationen i ett annat sökgränssnitt, utan få en mer användarvänlig upplevelse via en integrerad beställningsvy. Av denna anledning har användningen av Kohas OPAC som gränssnitt för låntagaren valts bort.

Fraplockningslappar

I Koha finns det fraplockningslistor men inte fraplockningslappar. Det har därför tagits fram en utskriftslösning efter GUB:s verksamhetsbehov. Lappar för respektive beställning skrivs ut direkt på ägande bibliotek oavsett om beställningen görs i Koha eller i beställningsvyn.

Mina lån och webbregistrering av nya låntagare

GUB använder Mina lån-sidorna som finns i Kohas OPAC, men anpassar dessa både till funktion och utseende. Inloggning via CAS har införts men för externa låntagare används lånekortsnummer och personnummer. Flera element har tagits bort eller bytts ut för att svara mot verksamhetens behov. Exempel på förändrad funktionalitet är anpassning av de informationsmeddelanden som visas om en låntagare har en spärr samt skapandet av ett nytt formulär för registrering av nya låntagare.⁸ GUB har också arbetat med den svenska översättningen av Mina lån mot Kohas översättningscommunity.

Anpassning personalgränssnitt

Det har gjorts en mängd anpassningar av webbgränssnittet i Koha utifrån verksamhetens önskemål. Hittills har det främst omfattat cirkulation och döljande av olika element på sidorna. Vissa element döljs helt och hållet, andra döljs utifrån parametrar baserat på rättigheter. Räkningansvariga kan till exempel se cirkulationshistorik och transaktionsloggar, medan personal inloggade i informationsdiskarna inte kan göra det.

Efter utredning och tester visade det sig att GUB:s läsesalslån inte kunde hanteras av Kohas funktion "Lån på plats". Situationen har lösts genom att istället använda Kohas utlåningsmeddelanden som

⁸ bibliotekskort.ub.gu.se

appliceras på läsesals- respektive institutionslån genom val i en drop down-menyn som är tillagd under utlåningsrutan.

Kohas användning av flera möjliga meddelandeinställningar har förenklats och funktionen anpassats så att endast ett meddelandeval styr alla Kohas typer av meddelandeutskick.

Förvärv och edifact-meddelanden

GUB har EDI-avtal med tre leverantörer vilket innebär framställning och automatisk överföring av förvärvsfiler. Koha hanterar dock inte en situation där leveransadresserna skiljer sig från besöksadresserna. Eftersom GUB har en centraliserad förvärvs- och katalogiseringsprocess så skickas nyinköpta böcker till en enhet medan fakturan skickas till beställande enhet. Ingen av dessa adresser motsvarar besöksadressen. En egenutvecklad plugin har därför tagits fram, vilken hjälper oss med åtskillnaden av adresser samt utskriften på edifact-beställningen.

Librisimport

När MARC-poster laddas in i den lokala katalogen efter katalogisering i Libris så behöver posterna åtgärdas på flera sätt: matchning på fält 035 måste ske, olika fält måste skyddas, exemplar skapas och data anpassas. Koha hanterar inte detta på det sätt som vår verksamhet önskar. För att åtgärda detta så har det utvecklats ny funktionalitet som appliceras i samband med Kohas egen importfunktion.

Elasticsearch

GUB valde att implementera den mer avancerade sökmotorn Elasticsearch istället för sökmotorn Zebra som sedan tidigare var integrerad i Koha.

Det har lagts en stor utvecklingsinsats på att föra integrationen av Elasticsearch vidare till en användbar nivå. Intresset från andra bibliotek i Kohas community har varit stort och GUB har nu tillgängliggjort utvecklingen tillsammans med instruktioner om hur det kan testas, med förhoppning om att det kan komma in i Kohas officiella kod så snart som möjligt.

Övriga integrationer

Förutom den redan nämnda integrationen med discoverysystemet Primo har även integrationer mellan Koha och andra system utvecklats och anpassats. Exempel på detta är utskriftssystem, grupprumsbokning, offentliga datorer, GU-kort och UB:s fjärrlånesystem. Se bilaga 2.

Flera av de integrationer som GUB skapade mellan Koha och andra system kunde inte testas i renodlade testmiljöer. Dessa integrationer fick istället testas när den första versionen av Kohas produktionsmiljö med migrerad data hade skapats. Produktägare för de system som berördes deltog i verifieringen av dessa integrationer.

Bokomaternas integration mot Koha kunde testas i en testmiljö eftersom projektet disponerade över en bokomat för testning. Dessa tester ledde bland annat till att bokomaternas utskrifter behövde anpassas för att fungera mot Koha. Eftersom testerna genomfördes på *en* bokomat upptäcktes inte att varje bokomat behöver ha tillgång till en egen SIP-session och att SIP-servern därmed måste tillåta flera sessioner. Detta fångades upp under testdagen, dagen innan Koha gick i drift, och kunde då åtgärdas.

En mer utförlig redogörelse för utvecklingsarbetet återfinns i bilaga 3.

Övriga anpassningar och förändringar

Det har skett en översyn av alla notifieringar och utskick så som e-post, sms och brev. Syftet har varit att skapa konsekvens och samstämmighet mellan språkbruket i systemets utskick och det språk som användaren möter i GUB:s övriga gränssnitt.

En liknande översyn av de benämningar som visar placeringen av bibliotekets material har genomförts. Det har varit viktigt att tillse att de är beskrivande och begripliga, då de visas i exemplarvisningen och i beställningsflödet.

GUB har arbetat med och kommer kontinuerligt att fortsätta arbeta med den svenska översättningen och val av formuleringar i Kohas personalgränssnitt.

Exempel på ytterligare förändringar är hur förkomna böcker faktureras samt hanteringen av långlån. En annan stor förändring är att GUB nu har frångått en separat OPAC och har discoverysystemet som enda sökgränssnitt.

Servermiljö och migrering

UB har två fysiska servrar för produktions-Koha. Den ena servern innehåller Koha med tillhörande databas, Elasticsearch samt Mina Lån och använder operativsystemet Debian, den andra servern innehåller de applikationer som utvecklats runt Koha, såsom beställningsvyn, bibliotekskort och GU-kort. Denna server använder operativsystemet Ubuntu. Orsaken till att det valdes fysiska servrar istället för virtuella var att säkerställa att Koha inte påverkas av andra system samt omvänt inte påverkar andra system. UB har också en server för Koha-staging som används för utbildning, verksamhetstester och liknande. Slutligen har utvecklarna en Koha-lab server där förändringar av Kohas källkod testas och där Kohas produktägare kan utföra tidiga tester av vissa förändringar.

GUB:s ILS innehåller en stor mängd data. Migreringen av data utvidgades och förbättrades succesivt för varje migreringsomgång, för att säkerställa kvaliteten.

Den första migreringen kördes i augusti 2016 och omfattade låntagare och böcker och den sista migreringen kördes i slutet av april 2018, i samband med driftsättning av det nya systemet. Samtliga migreringar, utom de sista, kördes mot Kohas olika testmiljöer där resultatet utvärderas. Den slutliga migreringen får anses ha varit av hög kvalitet, då den endast innehöll 11 migreringsfel.

Totalt gjordes drygt tio migreringar och den sista migreringen inför driftsättning gav följande antal poster:

- 1 619 887 bibliografiska poster
- 1 594 389 exemplarposter
- 64 412 auktoritetsposter
- 40 681 prenumerationer
- 96 257 låntagarposter
- 52 371 aktiva lån
- 4 575 aktiva reservationer
- 19 638 avgifter

För att säkerställa en god kvalitet i migreringen av data mellan Virtua och Koha genomfördes kontinuerliga tester utifrån processkartläggningen. I direkt anslutning till driftsättningen genomfördes också ett separat migreringstest med egna testfall. Detta test hade den svåra uppgiften att försöka hitta det som inte fanns, det vill säga försöka kontrollera om migreringen saknade någon data. Inga allvarliga problem hittades och den slutliga migreringen godkändes för produktion.

Förberedelser för implementation

Konfiguration av Koha

En omfattande del i att sätta upp Koha för produktion var konfigurationen av Kohas regelverk. Detta stora och komplexa arbete utfördes parallellt med att utveckling och därmed tillhörande återkommande tester av Koha genomfördes. Kunskapen om regelverket på GUB byggdes därmed successivt upp i ett samarbete mellan många inblandade.

Behörigheter i Koha

GUB använder ett antal olika behörigheter i Koha:

- "Superlibrarian" med full behörighet till systemets alla delar
- Förvärvsbibliotekarie, med full behörighet till förvärvsmodulen och budgethantering
- Fakturabibliotekarie, med full behörighet till transaktionsloggar och avgiftshantering
- "Diskinloggning", en gemensam inloggning för personal vid bibliotekens informationsdiskar, viss begränsad behörighet, exempelvis vad gäller förvärv och avgifter.
- Bibliotekarie, personlig inloggning med samma behörighet som diskinloggning, gäller all GUB:s personal som standardbehörighet
- Bokmater, en inloggning per bokomat, behörighet till in- och utlåning

Utbildning

Utbildningarna påbörjades i början av februari och pågick under sex veckor. Veckorna före och efter driftsättning genomfördes även ett antal repetitionstillfällen. Därefter har vidareutbildning skett på biblioteken utifrån de behov som uppstått efterhand. Under utbildningsperioden genomfördes korta dagliga möten där utvecklare tillsammans med systemansvariga bibliotekarier hanterade de olika frågor som uppkom i samband med utbildningarna.

De två första veckorna erbjöds fem tillfällen av mer allmängiltig karaktär och samtliga medarbetare var inbjudna. De kollegor som arbetat med utveckling och anpassning av Koha inledde med information om det nya systemet och en genomgång av de förändringar som byte av system medfört. Därefter presenterades och exemplifierades processerna sök, cirkulation, förvärv, katalog samt periodika.

Därefter följde fyra veckor med fördjupad utbildning inom processerna:

- Sök - 6 tillfällen
- Cirkulation - 7 tillfällen
- Katalog - 4 tillfällen
- Periodika - 2 tillfällen

- Förvärv - 3 tillfällen

Till varje process utarbetades ett material som också återfinns tillsammans med andra Kohabiblioteks lokala manualer på Svenska Kohanätverkets wiki.⁹ Utbildningstillfällena var två timmar långa och varvade teori och praktiska övningar. Utbildningarna genomfördes av processägarna, projektets testgrupper samt ytterligare medarbetare som arbetat med utbildningsmaterialet. I stort sett samtliga medarbetare deltog i flera utbildningar under veckorna före driftsättning.

Informationsplan inför systembytet

Inför nedstängningen av Virtua och vid övergången till Koha genomfördes en informationsinsats i tre steg:

- En uppmaning till studenter och GU anställda att planera sina ärenden på biblioteket annonserades på webben samt via affischer på samtliga bibliotek.
- Via webb och e-post informerades bibliotekets användare om att bibliotekssystemet var nedstängt. Bibliotekets lokaler var dock tillgängliga under tiden systemet var stängt.
- Information om att det nya systemet var i gång och vilka förändringar detta innebar annonserades på webben.

Detaljplanering av systembytet

Med utgångspunkt i schemalagda körningar och skript i systemet Virtua utarbetades en detaljerad nedsläkningsplan. Återlämningsdatum som inföll under de dagar systemet var stängt ändrades till ett senare datum. GUB valde också att stänga av uppräkningsavgifter från och med tidpunkten för systembytet till vårterminens slut. Övriga delar av Virtua kördes som vanligt fram till systemet stängdes ner.

Några dagar innan bytet inleddes gjordes den sista avgiftskörningen i Virtua och uppräkningsavgifter stängdes av.

Systembytet genomfördes enligt följande schema:

Dag 1 stängdes Virtua ned klockan 16.00. Data till sista migreringen plockades ut och därefter gjordes en extra backup

Dag 2 kördes själva migreringen in i Kohas produktionsmiljö

Dag 3 migreringen till produktionsmiljön verifierades och godkändes

Dag 4-5 var reserverade för att åtgärda eventuella fel, men inga åtgärder behövdes.

Dag 6 öppnades Koha för tester i produktionsmiljön, för hela GUB. Varje enhet genomförde tester enligt ett utarbetat testprotokoll och därefter gavs klartecken till styrgruppen om att öppna Koha kommande dag. Efter testerna återställdes databasens innehåll och en omindexering gjordes.

Dag 7 klockan 8.00 öppnades Koha för produktion

Under två veckor efter systembytet fortsatte de korta dagliga möten som inleddes under utbildningsperioden. Vid dessa möten fångades både akuta och mindre brådskande ärenden upp och

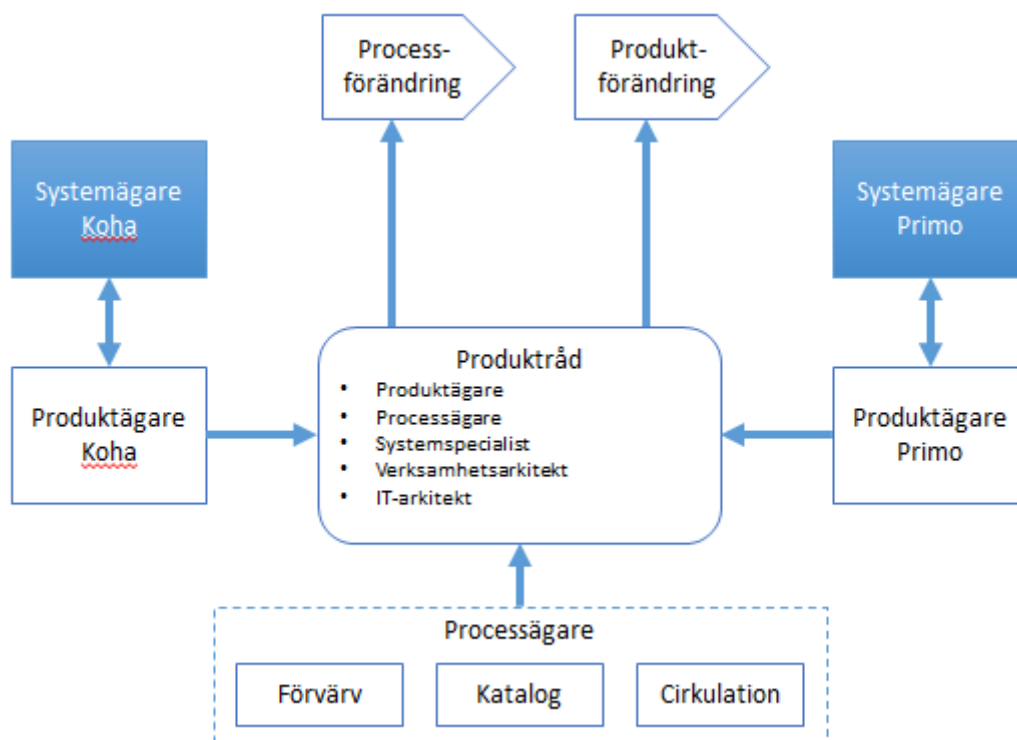
⁹ <http://koha.se/wiki/Dokumentation>

åtgärder kunde planeras utifrån grad av angelägenhet och verksamhetens behov. Återkoppling till hela GUB om rapporterade och åtgärdade problem skedde kontinuerligt.

Förvaltning av Koha och Primo

De två systemen Primo och Koha har ett stort beroende till varandra och bör därför ha en gemensam förvaltning. Förslaget är därför att ett gemensamt produktråd skapas där bland annat båda systemens produktägare ingår. Produktrådet kan på ett effektivt sätt samordna och hantera prioriteringar. Samordningen underlättas också av att båda systemen har samma systemägare på GUB.

Det pågår en översyn av GUB:s systemförvaltning som kan komma att påverka förvaltningen av Primo och Koha.



Vad har vi lärt oss?

NYS-projektet avslutades med att projektets deltagare utvärderade projektet genom att sammanställa sina erfarenheter under rubrikerna: vad gjorde vi bra, vad kunde vi gjort bättre samt överraskningar. Ett urval av de synpunkter som framkom får avsluta rapporten.

Vad gjorde vi bra?

- Bra manualer och välplanerad och riklig utbildning av personalen
- God förankring av Koha bland personalen, många kollegor stor grad av delaktighet
- Förmedlade positiv bild till övrig personal, goda insatser vände skeptiker
- Beslutet att skjuta fram Koha Go Live till påsken 2018 var ett viktigt beslut för projektets framgång, vi fick mer tid!
- Efter uppskjutet Go Live fanns tid att fokusera på rätt utveckling
- Tydlig uppdelning i vad som måste vara klart för att kunna Go Live och vad som kunde vänta
- Kartläggning av processer som sedan utgjorde grunden för utvecklings/testarbetet i projektet
- Att UB börjar tänka processer
- Prioriteringarna och förberedelserna inför Go Live, och att det fanns tid för att förbereda och stabilisera Koha på GUB och att anpassa delar av Koha till GUB:s verksamhet.
- Den iterativa utvecklingen av Koha-tillägg och förbättringar samt migreringen har fungerat mycket bra med ett team med mycket kunniga och duktiga utvecklare och produktägare.
- Inriktningen mot open-source har fungerat mycket bra i samband med GUB:s förändringar och bidrag inom Kohas open-source nätverk.
- Utvecklingens framtagna deploy-rutiner för att på ett säkert sätt kunna ta in Kohas gemensamma utveckling och de lokala GUB förändringarna.
- Tydligt fokus på två samverkande system = ökad kunskap samt utvecklade samarbetsformer
- Samarbetet mellan utveckling och verksamhet har fungerat mycket bra och arbetsformer har utvecklats bland annat med fint stöttande av varandra.
- Testgruppernas arbete under hela projektet var fantastiskt bra och genomfördes av mycket kunniga och engagerade projektmedlemmar.
- Anlitande av extern konsult inför projektets Go Live del var bra.
- Koha gick i produktion på ett mycket bra sätt och påverkade inte den dagliga biblioteksverksamheten.
- I samband med Go Live så gavs mycket snabb respons på uppkomna frågor.
- Omfamnade vår okunskap

Vad kunde vi gjort bättre?

- Den initiala planeringen var alltför snäv tidsmässigt och den tidiga dialogen kring detta mellan projektgrupp och styrgrupp kunde varit öppnare.
- Processerna vidareutvecklades inte under projektiden
- Mer tid att förenkla och ifrågasätta regler och processer
- Tydligare klargjort att önskemål om ny funktionalitet sannolikt inte prioriteras högst
- Mer transparent backlog inklusive avstämning av denna med processägare.
- Glömde lite bort Primo i utbildningen, hade varit bra med en användarstudie inför att Primo blev katalog
- Webbteamet skulle ha deltagit tidigare i projektarbetet
- Arbeta bort mer verksamhetsskuld
- Mer "Energi och glädje i projektet, WOW-känsla"
- Ett riktigt projektrum hade varit värdefullt och ökat projektets känsla.
- Mer och tätare information ut från projektet till hela GUB, den var alltför sporadisk.
- Kontakt processägare och EÖG-grupper kunde varit bättre och tätare
- Dokumentation och skriva på slutrapporten under projektets gång

Överraskningar?

- Att vi kunde lösa problem vid Go Live så snabbt.
- Att problem uppstod med rfid-läsare och bokomater.
- Mängden arbete med migrering och sent lyft av vissa migreringsproblem.
- Att slutresultatet blev så bra – "Nästan perfekt".
- Att så många kollegor är positiva – "Koha är bättre än väntat"
- Hur pass lite kollegor hittills använt Primo
- Stor hjälp från extern konsult - planering, projekterfarenhet, pepp
- Att systembytena gick helt enligt planering.

- Mängden arbete med att integrera Elastic search.
- Andra europeiska biblioteks stora intresse för Elastic Search.
- Att PostNord-Strålfors har agerat så långsamt inför GUB:s önskan att brevkommunikation med låntagare skall ske med hjälp av deras tjänster.
- Kohas hantering och konfiguration av förseningsavgifter skapade mycket utredningsarbete.
- Att onsite checkout inte fungerade för läsesalslån.
- "Att alla på UB är så trevliga"

Bilaga 1: Projektbemanning

NYS projektgrupp

Birgitta Andréasson	Humanistiska biblioteken
Cecilia Alfredsson	E-resursteamet
Emma Ternrud (tidigare Gustafsson)	E-resursteamet
Helen Gravås	Utvecklarteamet
Ingela Fryklund	E-resursteamet
Linda Hammarbäck	Biomedicinska biblioteket
Magnus Kull	Webbteamet
Monica Tengström	E-resursteamet
Rebecka Gustafsson	E-resursteamet
Stefan Berndtsson	Utvecklarteamet
Susanna Schlehaider	Samhällsvetenskapliga biblioteken
Ulrika Fröberg	Samhällsvetenskapliga biblioteken

Extern projektledare

Roberto Bonet (- augusti 2016)	Consid
Sture Ågren (augusti 2016-)	Consid

Processägare

Isabel Folkesson	Katalog	(Humanistiska biblioteken)
Lena Larsson	Förvärv	(Humanistiska biblioteken)
Ulrika Fröberg	Cirkulation	(Samhällsvetenskapliga biblioteken)

Ett mycket stort arbete utfördes av medarbetare i:

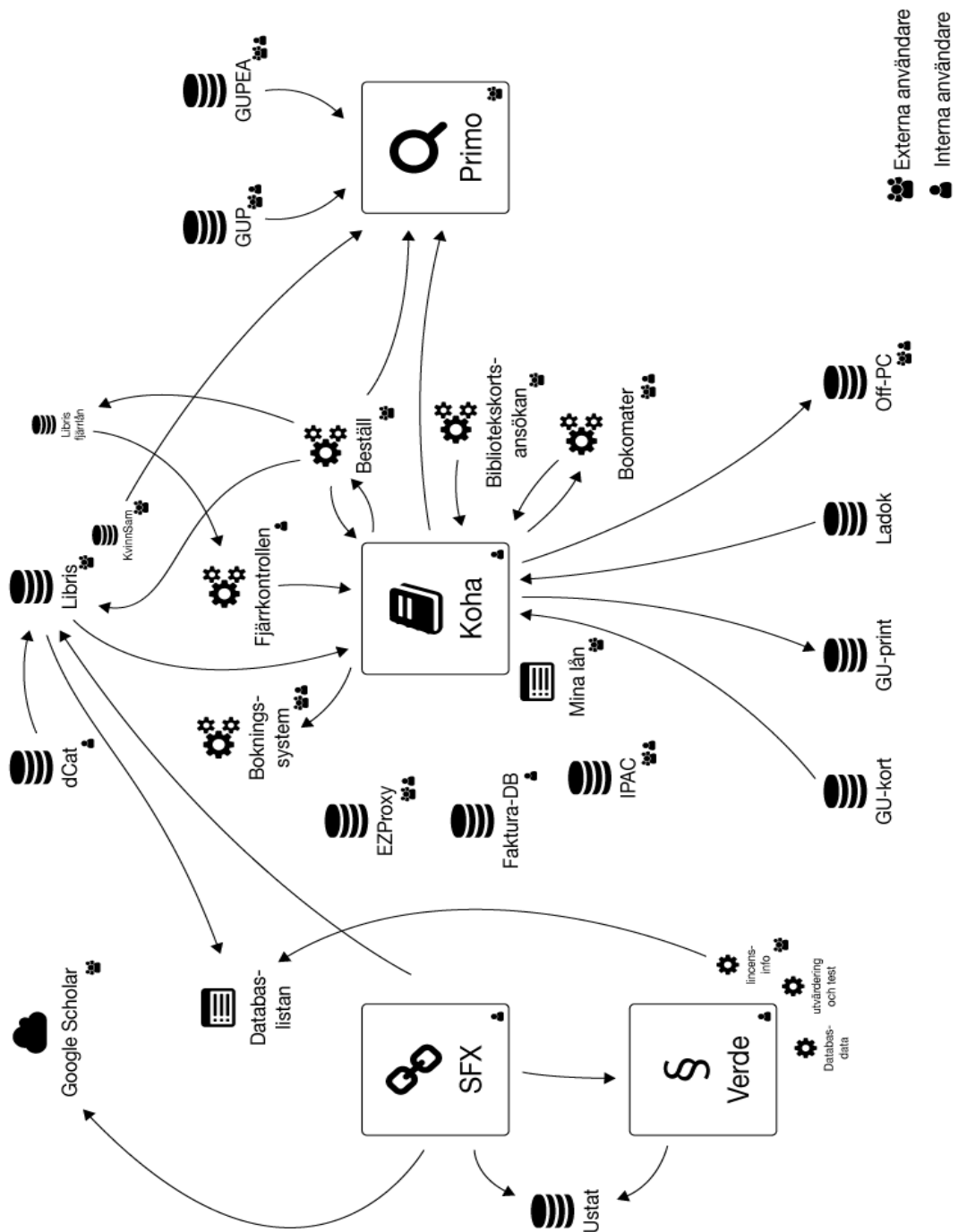
E-resursteamet
Utvecklarteamet
Webbteamet

Ett stort antal av GUB:s medarbetare deltog i testgrupper, framtagande av manualer och undervisningsmaterial, samt ansvarade för Kohautbildningarna.

Styrgrupp

Annelie Janred	Biomedicinska biblioteket
Ann-Mari Jonsson	Digitala tjänster
Camilla Gillén	Digitala tjänster
Helena Siesjö	Humanistiska biblioteken

Bilaga 2: Systemskiss



Bilaga 3: Teknisk dokumentation

Utveckling av Koha och kringssystem

Koha som Open-Source-system var en faktor till att välja denna produkt. Då vi har utvecklingskompetens inom biblioteket var vi beredda på att både kunna och behöva göra åtgärder i och kring Koha. Huvudfokus har varit att få till bra och konsekventa gränssnitt mot låntagarna, samt att implementera stödfunktionalitet för personalen.

Produkten Koha används främst av personalen inom biblioteket. Låntagarna möts endast av en väldigt nedbantad variant av OPAC-sidorna. Övrig interaktion med låntagare sker via system som bara får data från Koha.

Låntagarna söker och beställer aldrig i OPAC-vyn i Koha, utan det sker via externa system. Vi har därför delat upp vår utveckling i lite olika delar för att samverka mellan Koha och dessa externa system.

Koha

Utveckling i Koha sker med språket Perl och i viss utsträckning Javascript för gränssnittsrelaterade delar. Vår målsättning är hela tiden att det vi utvecklar i Koha ska lämnas över till Koha-Community så att andra kan ta del av resultatet. Vissa undantag finns för funktionalitet som är högst intern för oss, eller som bara är tillfällig.

ElasticSearch

En mycket stor del av utvecklingen har baserats på att vi valt att använda ElasticSearch som sökmotor. Denna sökmotor tillhandahålls av Koha, men är ännu inte den primära. Det man får från en okonfigurerad Koha är en sökmotor som heter Zebra. Vi insåg tidigt att denna sökmotor är gammal, svår att konfigurera, långsam och saknar en massa funktionalitet så som bra facettering. Därför valde vi att använda ElasticSearch. Koha-Community önskar också en möjlighet att helt övergår till ElasticSearch.

Tyvärr har ElasticSearch inte varit färdig nog för att användas, vilket innebär att vi lagt mycket tid på att få den i ett skick som var acceptabelt för oss, och som också kan hjälpa Koha-Community med en övergång.

Det här betyder att vi behövt se till att fältnamn synkroniserats mellan Zebra-konfiguration och Elastic. Ett indexeringsverktyg (Catmandu) som Koha-Community valt för att leverera data från MARCXML till Elastic visade sig vara väldigt långsamt i vissa steg och har därför bytts ut till en enklare variant. Till exempel tog det ca 3 sek extra varje gång en bok lånades ut eller återlämnades. Det skapade inte bara irritation hos personal som skulle återlämna många böcker, utan orsakade också problem i bokomater som såg ut som de hängt sig en kort stund för varje lån. Efter bytet av Catmandu upphörde dessa problem.

Ytterligare uppsnabbning av indexering genom att effektivera inläsning av data ur databasen har också implementerats.

Enklare viktning av sökresultat har lagts till, och ett arbete har påbörjats för att göra en modell där viktning är konfigurerbar av administratörer i Koha.

En plugin har skapats och en hook för att anropa den, precis innan indexering, för vi vill kunna indexera några få saker som inte bara beror på innehållet i ett enstaka fält, utan är en kombination

av vad som står i flera olika. Denna plugin kollar av kombinationen, och skapar ett nytt fält som sedan är det som indexeras.

En facett har också lagts till för att visa detta nya fält.

Gränssnitt

För personalens räkning har vi gjort många olika förändringar i både gränssnitt och funktionalitet. En del är gjort med systempreferenser, och en del är gjort med javascript/jquery.

Av säkerhetsskäl vill vi inte låta all personal komma åt cirkulations- och reservationshistorik. De har därför dolts så att endast den lilla grupp personer som kommer åt ändringsloggen får komma åt dessa två historiker.

Det sker en hel del manuella import av MARC-poster via gränssnittet. Här har vi gjort några förenklingar genom att gömma fält vi inte använder, förfylla en del, och bara visa import-paket som man själv laddat upp.

Förvärvsdelarna har förenklats en del med tydligare val i vissa delar, och reducering av listor, så att man inte kan välja fel konto när man ska skicka en order.

Då våra exemplars "shelving location" (LOC) hör ihop med biblioteket det står på, så finns nu en funktion där ett val av LOC automatiskt väljer rätt bibliotek vilket minskar risken för fel.

På några ställen har vissa knappar blivit standardval så att det räcker med ett Enter-tryck för att komma vidare.

Vi har försökt behålla funktionstangenterna så långt det varit möjligt, för att göra det enklare för personalen att göra "som de alltid gjort". Funktionstangenterna styr också av-/påarmning av böckerna, vilket gör det smidigt att jobba med.

Koha har normalt ett fritextfält där man skriver in varför en låntagare är begränsad. Vi har ett fåtal standardvarianter här, så istället för fritext kan man nu välja en dropdown med alternativ, där ett par alternativ också har utrymme för ytterligare text.

Som nämns under "Mina lån" nedan så har vi en förenklad valruta för meddelandeutskick. Personalen kan göra valet "Papper" också, vilket inte låntagarna kan göra själva, men i övrigt fungerar de likadant.

Avancerad sökning har reducerats och modifierats ganska mycket. Dels har några direkta buggar fixats, men främst har det lagts till alternativ som passar våra önskemål inom biblioteket.

Om man gör en enkel sökning i katalogen via sökrutan överst så behålls söktermen när resultatlistan visas.

En del uppsnabbning av sökresultat och detaljvyer när det finns många exemplar har också gjorts. Det ställs en mängd frågor till Elasticsearch i onödan i de här fallen, så vi har sett till att det bara görs om man faktiskt ska använda resultatet av dem.

Funktionalitetsändringar

Meddelanden som skickas när en bok krävs in (Overdue) skickas med en metod som sätts baserat på låntagarkategorin, inte den individuella låntagarens önskemål. Vi har lagt till en inställning där man kan ändra detta så att låntagarens val gäller även för denna sorts meddelanden.

Referenslitteratur lånas vanligen inte ut, men ibland sker det hos oss. När man då återlämnar boken vill man inte att den ska trigga en reservation, därför har en inställning lagts till för att stoppa detta.

Koha använder MARC-fältet 999c för bib-id. I de flesta andra fall används 001 för detta. I samband med migrationen såg vi till att alla bib-id hamnade i både 001 och 999c. Sedan har vi lagt till en inställning så att Koha automatiskt sätter 001 i samband med att den skriver 999c. Det här förenklade mycket i exporten till vårt discoveryssystem.

Vi är lite snällare än normalbeteendet hos Koha när det gäller regelverket kring när en förseningsspärr tas bort när en bok återlämnas. Koha tar bort spärren om inga andra förseningar finns, men vi tillåter andra förseningar så länge de inte är så sena att de också skulle givit upphov till en spärr.

Det finns många sorters lån hos oss. Både interna och hemlån. Vi vill egentligen behandla de flesta lån som helt normala, men bara veta vilken typ de är markerade att vara. Vissa böcker får man bara låna inom biblioteket. Onsite-checkout-funktionen som finns med i Koha upphäver vissa regelverk som vi egentligen vill ha (omlån t ex), därför har vi valt att inte använda den funktionen för det här.

Det gjordes en patch till Koha-Community för att skydda fält i bib-poster när man sparar dem. Vi har fixat till denna patch så att den fungerar lite bättre och enklare och är uppdaterad med aktuell Koha-kod. Den används t ex för att fält som importeras från Libris inte ska skriva sönder vissa av oss lokala fält.

Vår leverans av SMS-utskick sker genom att vi skickar epost enligt ett visst format som sedan blir ett SMS till låntagaren. Vi måste dock skicka från en specifik avsändaradress, så där fick vi lägga till en inställning för att tvinga fram detta.

I samband med detta vill vi inte heller att meddelanden om att bok finns att hämta ska skickas direkt, personalen ska få lite tid att faktiskt lägga boken på hyllan. Vi har därmed lagt till en fördröjningsparameter till process_message_queue.pl så att vi kan fördröja utskick ett tag.

Förvärv sker i huvudsak via EDifact. Det stöd som finns för EDifact i Koha saknar möjlighet att lägga in en leveransadress separat från faktureringsadress. En plugin lägger därför till det baserat på plugin-inställningar och val i förvärvsmodulen.

EDifact-meddelandena exporteras från Koha varje kväll och skickas till leverantör en gång per dygn.

Det finns några konstigheter i hur prenumerationer hanteras. Det ställs i vissa sammanhang väldigt många frågor till databasen, samma fråga ställs massor av gånger. En liten cache har lagts till för att frågorna inte ska lasta ner databasen.

Mina lån

En vy vi kallar "Mina lån" har implementerats på ett liknande sätt som Luleå universitetsbibliotek valde att göra. Vi använder OPAC-sidorna i Koha, men vi har plockat bort alla sökrutor, kö/reservation, låntagarregistrering med mera och visar bara saker som rör låntagarens egna lån, reservationer, avgifter, personuppgifter och önskemål om hur meddelanden ska skickas.

För att förenkla valet för skickande av meddelanden har vi utvecklat en förenklad valruta. Låntagaren kan inte välja olika utskicksmetoder för olika typer av meddelanden, utan kan bara välja mellan e-post, SMS eller e-post+SMS.

Vi har också snyggt till visning av spärrar och andra begränsningar, så att låntagaren får en enkel lista på vilka begränsningar de drabbats av, och vissa länkar till var de kan hitta mer information.

När låntagaren ändrat sina uppgifter har vi lagt till en funktion som automatiskt godkänner ändringen, så att personalen inte behöver bekräfta det varje gång.

Låntagarregistrering

GU-kort

Information om studenter, och till viss del personal, inom GU får vi via en integration med IT-enheten på GU. Vi har därför en sidoapplikation som tar emot data från GU och skapar låntagare i Koha, eller uppdaterar befintliga låntagare. Vi får också information om låntagaren spärerar sitt GU-kort centralt, så att vi kan spärra i Koha strax efter.

Denna applikation är skriven i Ruby on Rails.

Bibliotekskort

De personer vi inte får data om från GU kan registrera sig själva via en annan extern applikation. Även denna är skriven i Ruby on Rails, men den har också ett javascriptgränssnitt skrivet i EmberJS. Detta ger oss lite mer flexibilitet kring vilka fält som visas, hur de visas och regelverk kring dem som vi hade svårt att uppnå i OPAC-vyn i Koha.

Både denna och GU-kort-tjänsten har ett par Perl-script som ligger på Koha-servern för att hämta och skapa/uppdatera låntagarposterna direkt i Koha. Vi skriver inte direkt till databasen, utan använder de Perl-funktioner som Koha tillhandahåller för ändamålet.

Bokomater

Det mesta kring bokomaterna fungerar bra i Koha, men vi har behövt göra några tillägg och förändringar. Normalt får man ägande bibliotek som sorteringsvärde från Koha via SIP2-protokollet. Vi har lagt till en funktion så att man istället kan välja (via systempreferens) att få "shelving location" (LOC) som för oss ger en bättre finmaskighet i hur vi kan sortera i våra återlämningsautomater. Vi har en inställning så att man inte ska få en Alert-signal om man återlämnar en bok som inte var utlånad, då det ställde till det lite med sorteringen.

Om någon står på kö på en bok när den återlämnas i en automat så kommer Koha normalt att fånga upp denna kö och generera ett brev till personen att boken nu finns att hämta. Detta var lite olyckligt eftersom boken kanske blir liggande i automaten och inte alls finns att hämta ännu. Därför har vi en inställning som kan säga till bokomatrutinen att inte fånga upp denna kö, utan bara sortera boken på att det finns kö. När sedan personalen går igenom de böckerna fångas den upp och läggs på hyllan.

Slutligen har vi ett tillägg som säger att om en bok är reserverad (men inte uppfångad för en specifik person ännu) och någon annan än den som står i kö vill låna den så ska bokomaten tillåta detta. Vanligen blir man stoppad här, men vi vill ge företräde åt den som befinner sig på biblioteket. En bok som hamnat på avhämtningshylla och där låntagaren fått meddelande om boken får naturligtvis endast lånas av rätt person.

Autentisering

Bibliotekssystemet används för autentisering i flera olika externa tjänster, samt "Mina lån". Vi lagrar personnummer i ett "borrower attribute", och vill att låntagaren ska kunna använda sitt bibliotekskortsnummer samt personnummer för att logga in. I och med detta har vi lagt till en möjlighet att använda just denna kombination mot Koha, samt lagt till några Perl-script som externa tjänster kan använda för att fråga om. Exempel på detta är vår grupprumsbokning och inloggning för allmänheten på våra offentliga datorer.

Integrationer

GUPrint

Koha är källsystem till några andra tjänster inom verksamheten. Allmänhet ska kunna använda den utskriftstjänst GU har (GUPrint). För att det ska vara möjligt skickar Koha data till GUPrint när låntagarposterna uppdateras.

Fjärrkontrollen

Vårt egenutvecklade system för att hantera Fjärrlån, och som pratar med Libris Fjärrlån heter Fjärrkontrollen. Detta är ett relativt fristående system, men det har möjlighet att automatiskt skapa en bibliografisk post i Koha, samt ett tillhörande exemplar. Detta exemplar kan sedan lånas ut som vanligt till låntagaren. Vid återlämning kan Fjärrkontrollen radera posten och exemplaret ur Koha.

Som med övrig integration sker detta via Perl-script som anropar Perl-funktioner i Koha.

Discovery

Vi skickar också uppdaterade bibliografiska poster till vårt discoverysystem (Primo) två gånger per dygn. Primo är den primära plats där våra låntagare kan söka efter våra böcker. För att kunna skicka rätt poster har vi fått lägga till stöd för att utesluta maskerade poster, samt en bättre hantering av datum-intervall.

När man tittar på en bok i Primo får man status om exemplaren och om det är en tidskrift får man information om vilket bestånd vi har av denna. Informationen för att hämta detta kommer från en beställ-applikation, som i sin tur hämtar informationen via Perl-script i Koha. Från Primo kan man också påbörja en beställning av materialet, som sedan utförs i beställ-applikationen.

Libris

Vi hämtar poster varje natt från Libris. Dessa behandlas först av ett fristående program (Adjustlibris) enligt en uppsättning regler så att posterna blir redo för import in i Koha. Därefter används kommandot bulkmarcimport för att ladda in dem i Koha. För att inte få dubletter, och för att kunna skapa exemplar från Libris-posten automatiskt har vi en plugin som letar upp Libris bibid i Koha via ElasticSearch, och om det hittas uppdateras posten, annars skapas en ny. Det finns också ett regelverk kring när nya exemplar ska skapas som sköts av samma plugin. Vi har fått utöka och rätta en hel del fel och brister i bulkmarcimport för att detta ska fungera.

Visning av exemplarstatus i Libris webbsök sker på liknande sätt som i Primo med en fråga till vår beställ-applikation.

Beställ

Inom vår verksamhet har vi behov av ett komplicerat regelverk kring vem som får köa och beställa vilket material från och till olika platser. Vi skiljer på beställning och kö, där beställning är en begäran att personalen ska hämta en specifik bok åt låntagaren, reservera denna och lägga den på hylla för avhämtning. Det kan också innebära att den skickas från ett bibliotek till ett annat beroende på var låntagaren önskar hämta den. Ofta beror detta på att boken finns i ett slutet magasin så att det inte är möjligt för låntagaren att hämta själv. Kö är situationen när boken är utlånad.

Beställning sker alltid på ett explicit exemplar medan kö för det mesta sker på bibliografisk nivå, För tidskrifter läggs däremot en kö på exemplaret eftersom varje tidskriftsnummer är specifikt. Båda dessa sorters reservationer blir reservationsrader för låntagaren i Koha.

Det finns också en tredje sorts beställning som vi kallat beståndsbeställning. Alla tidskriftsnummer finns inte som exemplar, utan exemplaren skapas när någon vill ha dem. Där kan man med en

beståndsbeställning säga "jag vill ha nummer 3 av senaste årgången" i fritext, och så får personalen ta reda på om det går, och i så fall skapa exemplar och reservation i Koha manuellt. I detta fall skapar beställ-applikationen ingen reservationsrad, utan det är helt osynligt i Koha fram till dess personalen gjort sitt.

För att personalen ska veta vad de ska göra skrivs det ut framplockningslappar på skrivare hos det bibliotek som äger det beställda exemplaret. Denna utskrift baseras dels på information som beställ-applikationen har men också på information som Koha känner till. Därför har vi lagt till en plugin (och en tillhörande hook som anropar denna plugin) som gör lite extra jobb med att samla ihop data efter att en reservation skapats i Koha.

Då beställ-applikationen och exemplarvisningen i Primo i stort visar precis samma saker, så är beställ-applikationen den produkt som levererar informationen till Primo för visning.

Denna applikation är skriven i Ruby on Rails, med javascriptgränssnitt i EmberJS.

Praktikaliteter i utvecklingsarbetet

Koha är en Open Source-produkt som sköts av ett gemensamt community av utvecklare över hela världen. GUB är nu en aktiv del av detta community. Tanken var att vi skulle anpassa koden till att bete sig och göra precis det vi vill, men också få tillgång till communityts löpande utveckling av koden i form av buggfixar och nya features. Detta ställer stora krav på hur vi hanterar källkoden och det har blivit viktigt att hålla koll på vilka kodändringar som kommer från GUB och vilka som kommer från resten av communityt. Den grundläggande tanken var att de flesta av våra kodändringar på sikt skulle komma in som ett bidrag till resten av communityt, men att det viktigaste på kort sikt var att få upp ett system som skulle fungera för oss. Att genomföra detta i praktiken har varit minst sagt komplicerat och ledde till att vi utarbetade en speciell struktur.

Strukturering av källkoden

Först och främst såg vi till att göra en så kallad *fork*, en egen kopia av communityts Koha på Github: <https://github.com/ub-digit/Koha>. Git är ett verktyg för versionshantering av källkod och Github är en webbtjänst som erbjuder git-funktionalitet. Här ville vi alltså både få in resultatet av utvecklingsarbetet från communityt och förvara resultatet av våra egna utvecklingsinsatser. Syftet var inte att uppnå en väsensskild produkt som skulle divergera från communityts Koha, utan snarare att kunna hålla koll på våra ändringar och att löpande berika vår kod. Det gick till så att master-branchen i GUB:s fork med jämna mellanrum synkroniserades med communityts master-branch.

Master-branchen

Denna innehöll endast kod från Koha-communityt. Med jämna mellanrum uppdaterade vi master så att den speglade communityts master. Vi lade aldrig någon egen kod här, utan det var alltid helt och hållet ren Koha-community-kod.

Bugg-brancher

Här sparade vi dels ändringar som syftade till att lösa buggar i communityts kod och dels kod tillhörande våra egna kodförbättringsförslag som vi föreslagit för Koha-communityt. Dessa skickade vi så småningom som patchar till communityt där de hanterades inom ramen för den normala bugghanteringsprocessen. Tanken med dessa var att dela med oss av lösningarna till communityt för att på så sätt bidra med förbättringar som skulle komma världen till godo. När en patch väl blivit accepterad och upptagen i master-koden, hade bugg-branchen fyllt sitt syfte och kunde tas bort.

Feature-brancher

Här hade vi våra egna ändringar som framför allt handlar om att få till något så som vi vill att det skulle fungera hos oss, med val och utseende för att passa vår lokala miljö. Exempelvis lät vi allt som hade med det vi kallar *Mina lån* att göra finnas i en egen branch.

Release-brancher

När vi var redo att driftsätta en viss funktionalitet såg vi till att skapa en ny branch med utgångspunkt från en commit-punkt i master-branchen. Därefter såg vi till att applicera varje feature-branch och varje bugg-branch på denna med hjälp av git rebase-kommandot. Under processen löste vi konflikterna som uppstod. Vi använde i detta ett verktyg som heter rerere som spelade in konfliktlösningarna, så att vi kunde återanvända dem vid senare release-byggen. Detta var en enorm tidssparare eftersom vi annars hade fått lösa samma kodkonflikter om och om igen. När sedan alla konflikter var lösta var resultatet en release-branch där alla våra ackumulerade ändringar av Koha lagts till Koha-communityts kod. Denna release-branch kunde sedan installeras på våra servrar.

Servermiljöer

Som hjälp under arbetet hade vi tre olika servermiljöer för olika ändamål. Först utvecklades och testades koden lokalt på utvecklarens laptop. Därefter samlades en grupp ändringar ihop till en release som installerades på en lab-miljö, där resultatet granskades av produktägaren för GUB:s Koha. Om koden klarade acceptanskriterierna samlades den ihop med andra uppdateringar och gick till slut vidare till en staging-miljö. Där testades uppdateringarna av en testgrupp som testade olika scenarier mera i detalj. Kod som passerat testningen på staging kunde därefter driftsättas till produktionsmiljön, vid lämpligt tillfälle och efter klartecken från produktägaren. Som verktyg för att driftsätta användes *Capistrano*, ett ruby-baserat verktyg som i olika steg såg till att koden uppdaterades, databasen uppdaterades, konfiguration laddades in i databasen och språkfilerna laddades och de olika cacherna startades om.

Pluginhanteringen

GUB har med gub-plugin-extender utökat möjligheten att använda plugins i Koha. Det går till på så sätt att på ett ställe i källkoden anges en hook. Om man har en plugin installerad som implementerar denna hook körs koden i pluginen, men om man inte har någon plugin installerad, händer ingenting. Denna variant av hook-baserad plugin returnerar samma datastruktur som den tar emot, men returen kan innehålla modifierade eller behandlade data, eller så kan vissa sidospår initieras inifrån plugin-koden. En plugin enligt denna modell körs automatiskt. Den behöver alltså inte startas manuellt eller väljas i något gränssnitt.