

Risico aanpak bij overdracht legacy systemen bij de overheid

Lucien Duijkers (6 februari 2020)

Uit mijn accountancy- en belastingervaring weet ik dat elk probleem handmatig verwerkt wordt.

Vanuit de wetten ga je stap voor stap naar de uitvoeringsregel en de jurisprudentie.

Dit complex geheel van regels is niet naar eenvoudige usecases te converteren.

Bij een functioneel beheergroep bij de overheid leerde ik de problemen bij de overdracht van programma's en het installeren op een nieuwe omgeving.

De programma's waren moeilijk te integreren met bestaande systemen door de veelvoud van gebruikte technieken.

Ik was de enige die de oude methodieken beheerste en het leuk vond om het te combineren.

Ik vond een veelvoud van leveranciers van Amerikaanse producten, zoals IBM, Oracle en Java

Ik heb me verdiept in de US ICT periode 2000-2010 waar de Oracle en IBM-producten zijn ontwikkeld

Ik zoek naar een oplossing voor de verscheidenheid van technieken wat leidt tot handwerk en verdiepen in een veelvoud van technieken

Ik vond de agile manifesto groep die een oplossing voor de integratie uit de telecommunicatie met veel leveranciers heeft gevonden.

Hun ervaring is dat het onder onderhoud brengen van bestaande code meer kans tot slagen heeft dan nieuwbouw.

Door de bestaande code transparant en zelf documenterend te maken is het werk voor beheer en onderhoud een stuk gemakkelijker te maken.

Lever aan het einde van project alle informatie op, die nodig is in de verdere keten.

Ik heb voor accountants een werkprogramma voor controle in leveranciers software opgezet.

Daarom mijn idee om een werkprogramma te maken voor de risico aanpak bij overdracht legacy systemen bij de overheid.

Risico	Beheer doelstelling	Mogelijke Beheersmaatregel
--------	---------------------	----------------------------

Beheer

Beheer krijgt onbekend product en techniek aangeleverd	Bekend zijn met de globale opbouw van applicaties	Aan het einde van de projecten het gerealiseerde plaatje opleveren met gebruikte technieken en een technisch ontwerp. Functioneel beheer betrekken bij het scrum teamoverleg
Niet uniform instellen van environment	Een plaats voor de omgeving variabelen.	Er moet een centraal tool komen voor de instellingen van applicaties.
Calamiteiten analyse	Snel overzicht bij calamiteiten	Er moeten beheer schermen zijn met toegang tot relevante instellingen en logfiles
De logging van programma is verspreid en niet toegankelijk voor beheer	Beheerders moeten snel probleemanalyse kunnen doen	Eenduidige logging over alle producten heen
Ontwikkel en test omgeving zijn verschillend	Ontwikkelaars werken voor productie	Standaard inrichting van systemen. De Omgevingen variabelen snel te vergelijken
Niet reproduceerbaar productieprobleem	Continuïteit systeem	Zorg dat er steeds een up to date productie like test omgeving is en wijst de inrichting toe aan persoon of groep
Verlies kennis van oude systemen	Het beheer belang moet geborgd worden	Kennis borging binnen overheid ivm overgang leveranciers

Onderhoud

Opstarttijd van een project is te hoog	Inwerktijd op een project moet beperkt zijn	Check op volledigheid van oplever en installatie informatie
Ontbreken start informatie bij onderhoud	Check op totaal oplever pakket inclusief alle sources en ontwikkelomgeving	Standaard oplever documentatie, sources, testplan en testbestand
Herinrichting project	De overnemende partij snel inzicht in werk en denkwijze van ontwikkelaard	Zorg voor overzicht communicatie binnen programma's, overzicht herstart procedure installatie procedures Welke gebruikte producten Hoe stel je monitoring in
Te lage of geen test coverage	Zekerheid over de werking van het product voor en na aanpassing	Opdeling in meerdere lagen programma. Bijvoorbeeld los te testen zijn van business layers, database en webservices.
Geïsoleerde te verschillende techniek	Integratiebeleid van gebruikte technieken met bestaande technieken	Conformereren wereldwijde standaarden voor clean code en onder houdbaarheid Zorg voor een zorgvuldige scheiding tussen oude en nieuw techniek en conversie Implementeer een anti corruption layer pattern
Gebrek kennis over het inrichten van een test omgeving	Stel realistisch eisen test omgevingen op.	Inrichting documententen en toegangs rechten
Testsystemen sluiten niet bij de productiesystemen	Uitval door niet ingecalculeerde omgeving issues	Leg verantwoordelijk bij ontwikkel leverancier over beheer productie omgeving
Onder houdbaarheid over hele functioneel levensduur	Levenslang gebruik systeem	Ontwikkel een methodiek voor langere tijd dat onderhoud mogelijk maakt. Een voorbeeld Clean code en software craftsmanship

Oude technische kennis verloren	Continuïteit van het bedrijf	Als de gebruikte techniek te specifiek en uniek, neem dan zelf de verantwoordelijkheid van opleiden personeel.
---------------------------------	------------------------------	--