

Nieuwe legacy bouwen

Iemand die zijn ziel stopt in het bouwen van software zal altijd denken: "Het systeem waar we nu aan beginnen wordt anders. Dit keer maken we het zoals het echt hoort, dit keer maken we geen fouten". We gebruiken de beste technologie (Java!), best practices, een strak ontwikkelproces. Maar telkens wanneer het project afgerond is, lijkt het weer mis te zijn gegaan.

Een dubbel gevoel blijft over. Trots op het harde werk, de vernieuwende aanpak, de strijd die geleverd is. Maar ook teleurgesteld in de negatieve reacties op het systeem. Want zodra een systeem klaar is, zodra het projectteam ontbonden is, blijft er niet veel over behalve mopperende collega's, afkrakende concurrenten, klagende beheerders en dus negatieve opdrachtgevers.

Hoe positief eindgebruikers ook zijn, hoe goed doordacht de aanpak ook was, bij het eerstvolgende project wordt wéér gedacht: "de volgende keer moet het anders". Het mooie nieuwe systeem is per direct legacy geworden. Legacy van de ergste soort: nieuwe legacy. Want met legacy is het net als met werknemers: bij problemen moeten de nieuwste collega's er het eerste uit. Legacy van twintig jaar terug moet haast wel goed zijn. Legacy van twee maanden geleden kan niet anders dan troep zijn.

Natuurlijk maken we met z'n allen wel eens slechte software. Ik ben zelfs arrogant genoeg om te denken dat vele ontwikkelaars niet eens weten hoe ze echt goeie software moeten

maken. Maar ook mij kun je aanspreken op fouten die ik vast wel gemaakt heb. Soms doen we weloverwogen concessies aan de kwaliteit, wegen we ontwikkeltijd af tegen kwaliteitswensen. Maar de algemene teneur dat nieuwe software niet goed kan zijn, daar moet toch meer achter zitten.

Ik begin steeds meer te geloven dat het eeuwige wantrouwen tegen nieuwe software volledig ontstaat door één gegeven: de reusachtige kloof tussen beheer en ontwikkeling. Natuurlijk doen beheerders hun stinkende best om hun systemen in de lucht te houden. Maar bij elk knelpunt is er altijd één doodeenvoudige manier om alle kritiek op het beheer af te wimpelen: "Het systeem is niet goed". Ik heb nog nooit meegemaakt dat het enthousiasme van ontwikkelaars volledig gedeeld wordt door beheerders. Negatief gezegd: er is zelfs geen enkele reden om tevredenheid over een systeem te tonen. Daarmee gooit een beheerder namelijk zijn eigen glazen in. Een systeem onder protest accepteren levert later de mogelijkheid om nog wat te schipperen met de beloofde dienstverlening. De volledige onmogelijkheid om beheerders te betrekken bij het bouwen van systemen hoort daar ook bij. Betrokkenheid betekent instemming. Instemming zorgt voor aansprakelijkheid. Wie wil er nou aansprakelijk zijn?

Aan de andere kant zullen ontwikkelaars kritiekpunten altijd verbloemen, want anders zouden zij het zichzelf wel heel moeilijk maken.

Het probleem ligt aan beide kanten. Ik ken geen enkele ontwikkelaar die baat heeft bij de knip tussen applicatiebouw en applicatiebeheer. Ik ken vele ontwikkelaars die denken dat zij zelf geen goede beheerders zouden zijn. Waarschijnlijk hebben zij gelijk. Beheerders leggen de nadruk op het feit dat beheer héél anders is dan bouw. Zij zien wél baat bij deze knip. Dat beheer en bouw verschillen, hoeft nog niet te betekenen dat we het niet kunnen combineren.

Waar projecten langzaam het ontwikkelteam omzetten in een beheers-team is veel meer succes en tevredenheid te vinden. Bouwers worden betrokken bij het beheer (en worden geconfronteerd met onvoorziene problemen) en beheerders leren een systeem ook echt kennen. Wanneer beheerders klagen over zo'n product, dan zijn ze er zelf bij geweest. Ontwikkelaars en beheerders worden samen verantwoordelijk voor de beheersbaarheid van het systeem. Het systeem wordt er misschien niet beter van, maar we maken ineens geen legacy meer. We maken dan gewoon productieve gezonde systemen met tevreden gebruikers, opdrachtgevers, bouwers én beheerders.

Maar helaas. Deze visie past niet in het kader van modern software management waar verdeel-en-heers centraal staan.

*Daan Kalmeijer is docent consultant bij
CIBIT adviseurs | opleiders
(e-mail: daan@cibit.nl).*