



**HOOGLERAAR
TIJS VAN DER STORM
PLEIT VOOR
EMANCIPATIE VAN
DIGITAAL ERFGOED**

SOFTWARE VERDIENT EEN MUSEUM



ER ONTBREEKT IETS. WE HEBBEN SOFTWARE EN EEN FLORERENDE SOFTWARE-INDUSTRIE. OOK HEBBEN WE MUSEA VOOR VRIJWEL ALLES WAAR WE TROTS OP ZIJN. MAAR EEN SOFTWAREMUSEUM IS ER NIET. CWI-ONDERZOEKER EN RUG-HOOGLEERAAR TIJS VAN DER STORM WIL DAAR VERANDERING IN BRENGEN. "HET MUSEUM EMANCIPEERT SOFTWARE TOT VOLWAARDIG DIGITAAL ERFGOED."

door Rolf Zaal foto De Beeldredactie / Peter Strelitski

BIJ EEN SOFTWAREMUSEUM DENKT Tijs van der Storm zeker niet in de eerste plaats aan ingelijste scherm-afdrukken of lappen broncode. Nee, de CWI-onderzoeker en RUG-hoogleraar wil spannende software voor een breed publiek live tentoonstellen in een online 'museum'. Baanbrekende programma's moeten uitgevoerd kunnen worden. Bezoekers moeten de gelegenheid krijgen unieke source code in te zien en bijvoorbeeld een stukje code kunnen veranderen om zo te ontdekken hoe dat de werking van het programma beïnvloedt. Daartoe zet hij zijn schouders onder het bij elkaar brengen van partijen die het belang inzien van preservatie van het snel van ons af vliedend digitaal verleden.

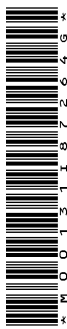
Een softwaremuseum; wat moeten we ons daarbij voorstellen?

"Een museum is een plek waar je naartoe kunt gaan om mooie en interessante dingen te bekijken. Die dingen zijn in dit geval programma's en de plek, die wordt uiteraard online. Laten zien betekent in het geval van software ook de mogelijkheid bieden om de software uit te proberen. De collectie zal natuurlijk vooral bestaan uit programma's die in hun tijd een doorbraak betekenden. Denk bijvoorbeeld aan eerste uitingsvormen van nu heel gewone toepassingen. Vrijwel iedereen heeft weleens met Excel ge-

TIJS VAN DER STORM is hoogleraar Software Engineering aan de Rijksuniversiteit Groningen en onderzoeker in de Software Analysis and Transformation (SWAT) groep bij het CWI in Amsterdam. Van der Storm houdt zich onder meer bezig met de ontwikkeling van domeinspecifieke programmeertalen (DSLs). Die maken het mogelijk om op een hoger abstractieniveau te programmeren, waardoor code eenvoudiger aanpasbaar wordt. Van der Storm is één van de ontwerpers van Rascal, een 'meta-compiler' voor het maken van DSLs.

werkt, maar wie van de gebruikers van nu realiseert zich dat Visicalc in 1979 voor het eerst het concept van spreadsheet neerzette? En hoe weinig er sindsdien eigenlijk aan is veranderd?

Word is mogelijk het meest gebruikte toepassingsprogramma in de wereld; maar wie kent Bravo nog (1974, Xerox PARC)? Dát was de eerste WYSIWYG-tekstverwerker en tevens een van de eerste programma's waarin het concept van elkaar overlappende vensters en muisbesturing voor het eerst werd toegepast. Bij een expositie van zulke doorbraakprogramma's hoort natuurlijk ook een audioguide, die programma's toelicht en hun opkomst, invloed en neergang technisch en economisch in een context plaatst. Verder heeft een museum vaak ook een kelder, waarin zaken te vinden zijn die het brede publiek minder aanspreken, maar die voor studenten en onderzoekers des te interessanter zijn. Denk bijvoorbeeld aan de broncode van een compiler als Algol 60 (1960). Die werd geschreven door Edsger Dijkstra, die daar een specifieke bedoeling mee had. Dat is op zich al een boeiend verhaal [hij wilde dat Algol de praktijk van programmeren zou veranderen], maar voor de geschiedschrijving van de technologie is misschien wel minstens zo interessant om te zien hoe Dijkstra bepaalde programmeertechnische problemen aanpakte. In de code van





“Software is niet minder belangrijk of interessant dan kunst, waarom handelen we daar dan niet naar?”

die Algol-60 compiler wordt bijvoorbeeld voor het eerste gebruik gemaakt van een runtime stack om recursive function calls te implementeren [stukken code die, als in een Droste-plaatje, gebruik maken van zichzelf]. Van dergelijke mijlpaalprogramma's bestaan kritieke edities, met annotaties en commentaren die de stand van het wetenschappelijke inzicht en de ontstaansgeschiedenis van en impact van zo'n programmacode weergeven. Voor iemand die het fenomeen software bestudeert, zijn de kritieke edities als wat de Gutenbergbijbel voor een linguïst is.”

Wat is het nut van een dergelijk museum?

“Dat het software emancipeert tot volwaardig digitaal erfgoed. Het museum doet recht aan het feit dat software – niet minder dan bijvoorbeeld ‘born digital’ kunst – een expressie van menselijke scheppingsdrang is. Je vraagt toch ook niet naar het nut van de Nachtwacht? Software als tentoongesteld artefact heeft zich vooralsnog onttrokken aan de traditionele digital heritage and history of computing disciplines. Met het museum of software hoop ik beide vakgebieden te overbruggen. Software is niet minder belangrijk of interessant dan kunst, waarom handelen we daar dan niet naar?”

Dat is het principiële argument. Wat

gaat er volgens u voor zorgen dat er ook bezoekers komen?

“Er is nu zo'n veel geciteerde kreet ‘software is eating the world’ [Marc Andreessen, 2011]. Als dat waar is, dan is het toch voor iedereen zaak om te begrijpen hoe dat zo kon komen? Ik stel me drie groepen museumbezoek voor. Geïnteresseerde leken die het leuk vinden om een paar keer een uur of zo bezig te zijn met de geschiedenis achter de software die hen nu omringt. Wat kon zo'n allereerste webbrowser, van Tim Berners Lee (Ervise 1992), nu wel en wat nog niet? Hoe zag het allereerste programma dat van een muis gebruik maakt (Bravo, 1974) er nu eigenlijk uit? Wat was er überhaupt voor de grafische interface er was? Misschien willen ze ook nog wel snappen wat een compiler doet.

De tweede groep bestaat uit professionals – niet per se IT'ers – die meer willen begrijpen van de ontwikkeling van software-engineering als discipline. Zij willen waarschijnlijk ook naar broncode kijken, zich verbazen over de wijze waarop in jaren zeventig werd gewoerd met geheugenruimte, willen snappen hoe de makers van Bravo toen nog ongekende programmeeruitdagingen rond de grafische gebruikersinterface oplosten. Als laatste groep zie ik de onderzoekers die kennis willen toevoegen aan wat we op dit moment over de geschiedenis van

software weten. Specialisten die broncode bewerken om het beter te begrijpen, er metrieken op loslaten. Dit is vergelijkbaar met onderzoekers die met een röntgenapparaat naar de Nachtwacht turen of mensen die willen snappen waarom iets in de Gutenbergbijbel zus en niet zo is geformuleerd.

Ik kan niet garanderen dat dat allemaal nut heeft in economisch opzicht, maar ik ben ervan overtuigd dat het wel zin heeft.”

Er bestaan al academische disciplines als ‘digitaal erfgoed’ en ‘history of computing’, waarom laat u als hoogleraar software engineering deze kwestie niet aan die collega's over?

“Dit is iets wat al heel lang door mijn hoofd speelt. Een belangrijk aspect is dat software een uniek, complex karakter heeft. Om een museum of software op te zetten moeten technische uitdagingen opgelost worden, en daar is software-engineeringexpertise voor nodig.”

Waar is het wachten nog op?

“Mijn plannen zijn nog erg pril en in ontwikkeling. Het is meer een intentie. Er moeten eerst nog heel wat weerbarstige kwesties worden opgelost, zowel organisatorisch als technisch. Om met de organisatorische uitdagingen te beginnen; er moet een gremium



ENKELE ORGANISATIES DIE HET INITIATIEF VAN TIJS VAN DER STORM STEUNEN:

- **Noterik, Rutger Rozendal:** "De opkomst van IT is misschien wel de meest ontwikkende trend van de afgelopen veertig jaar. Wij zien het tentoonstellen van de stapsgewijze veranderingen achter deze trend als een essentiële stap in de educatie van toekomstige generaties."
- **SIG, dr. Magiel Bruntink:** "SIG is zeer geïnteresseerd om deze grote werken voor een breed publiek tentoongesteld te zien om zo het begrip van en de kennis over software te vergroten."
- **UvA, Prof. Dr. Julia Noordegraaf:** "Het museum kan helpen de digitale geletterdheid naar het niveau te brengen waarop burgers op kritische wijze kunnen omgaan met de 'datafificatie' van onze maatschappij."
- **Digitaal Erfgoed Nederland, Marco Deniet:** "Archivering van software levert een belangrijke bijdrage aan het documenteren van de informatiesamenleving."

"Het museum doet recht aan het feit dat software een expressie van menselijke scheppingsdrang is."

komen dat het museum en de collectie beheert. Ik kan niet degene zijn die uit gaat maken wat de digitale pendant van de Nachtwacht is, of wat al dan niet goed genoeg is om als digitaal museumstuk te gelden. Daar moet een community of een stichting over gaan. Niet onbelangrijk is de funding, op dit moment krijg ik vooral bijval van organisaties die in de rol van gebruiker veel belang aan hechten aan de inrichting van zo'n museum. Heel belangrijk natuurlijk, maar ik heb nog geen mailtje gehad van een internationaal softwarebedrijf dat 2 miljoen euro beschikbaar stelt. Ik reken daar ook maar niet op en bereid me dus voor op veel passen en meten met diverse sponsors. Mogelijk komen we uit op een subsidie via de Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO), maar die eist wel cofinanciering vanuit bijvoorbeeld het bedrijfsleven. Ik ben druk aan het netwerken, maar de meest geïnteresseerde partijen tot nu toe hebben doorgaans het krapste budget. Waar met name kosten voor zullen moeten worden gemaakt – daarmee kom ik meteen op het tweede domein waar werk te verzetten is – is de technische realisatie van een site waarop allerhande software live is. De software die we willen laten zien is gemaakt voor hardware die zelf ook al in een museum thuishoort en het is geen optie om die 'on premise' operati-

oneel te hebben. Waar ik nu onder meer aan denk, is een generiek platform waarop virtuele varianten van de machines uit het verleden kunnen worden geïnstantiëerd. Je hebt het dan over software-emulatie van oude hardware. Het bouwen van zo'n software-defined computer is behoorlijk lastig en we zullen er diverse nodig hebben. Kortom, dat is een karwei dat je idealiter zou willen automatiseren; bouw een systeem dat specificaties van hardware als input accepteert en emulatiesoftware als output heeft. Dan is dat probleem voor eens en altijd opgelost. Het klinkt nogal ambitieus, en ik beloof ook niet dat het lukt, maar ik vind het het onderzoeken waard. Verder zal er – voor 'de kelder' van het museum – een op software toegespitst inventarissysteem moeten komen, waarin programma's, broncodes, commentaren, metadata, links, libraries en handboeken worden bewaard en ontsloten."

Heeft software esthetische kwaliteiten?

"Absoluut! Programmeurs spreken vaak van mooie of elegante code, of van conceptuele integriteit. Verder kan software van buitenaf natuurlijk ook mooi worden gevonden: het ontwerp van de user interface bijvoorbeeld of de graphics in een game. Kwaliteit is moeilijk te definiëren, maar esthetiek hier speelt mijns inziens zeker een rol bij." 