

Die Entwicklung innovativer Touch-Interface-Software für industrielle Anwendungen

Autoren:

Berner & Mattner Systemtechnik GmbH:

Dr. Klaus Wiltschi, Abteilungsleiter Industrial Customers

Dr. Michael Sturm, Abteilungsleiter Software Products

Centigrade GmbH:

Thomas Immich, Geschäftsführer

1. Zusammenfassung

Berührungsempfindliche Anzeigen – sogenannte Touch-Displays – bieten als moderne Mensch-Maschine-Schnittstelle (Human Machine Interface (HMI)) neue Möglichkeiten bei der Bedienung von Geräten und Maschinen in industriellen Anwendungen. Gelingt ein Interface-Design, das komplexe und umfangreiche Funktionalität intuitiv bedienbar macht, zahlt sich dies mehrfach aus. So lassen sich nicht nur Produkte mit durchdachten Bedienkonzepten über die erhöhte Attraktivität umsatzwirksam differenzieren und die Flexibilität für Variantenentwicklungen und Updates steigern, sondern gleichzeitig auch die Kosten für Bedienschulungen und Bedienfehler auf Kundenseite reduzieren. Entsprechend groß ist der Druck auf Hersteller, Produktmanager, Designer und Entwickler, derart optimierte Touch-Interfaces zu entwickeln.

Doch die Entwicklung innovativer Touch-Interfaces stellt die Hersteller vor neue Herausforderungen. So übertragen die Anwender ihre alltäglichen Nutzungsergebnisse aus Consumer-Geräten wie iPhone oder Tablet-PCs auch auf industrielle

White Paper

Anwendungen und bringen so neue Anforderungen an Design und intuitive Bedienbarkeit (Usability) ein. Anforderungen, die klassische Verfahren im Bereich Software-Entwicklung mit ihrer üblicherweise nachgeordneten Designphase für das Bedieninterface nicht erfüllen können.

Um moderne und attraktive Touch-Interfaces zuverlässig entwickeln zu können, sind grundlegende Änderungen im Software-Entwicklungsprozess erforderlich. Dieses White Paper stellt Methoden, Prozesse und Software-Entwicklungswerkzeuge vor, die sich in der Entwicklung preisgekrönter HMI-Projekte bewährt haben. Die wesentlichen Erfolgskomponenten dieses Ansatzes sind die frühe und intensive Beteiligung von Anwendern am Entwicklungsprozess, die parallel und agil verlaufende Entwicklung von Interface-Design und Software sowie der Aufbau einer Infrastruktur für die verteilte Entwicklung (Distributed Development Platform), um eine effiziente Arbeitsteilung von Spezialistenteams für Hardware-, Software- und Bedienoberflächen-Entwicklung zu erlauben.

2. Die Ausgangslage

a. Warum Touch-Interfaces?

Nicht nur im Consumer-Bereich haben sich berührungsempfindliche Anzeigen als Bedien-Interfaces durchgesetzt. Dominierten in industriellen Anwendungen früher Tastenfelder, Folientastaturen oder sogar externe Interfaces wie Keyboards oder Maus, so setzen sich auch hier verstärkt Touch-Interfaces durch.

Die Treiber hierfür sind die wirtschaftlichen Vorteile für die Hersteller von Geräten, Maschinen und Anlagen:

- Ihre Verbreitung in der Consumer-Elektronik hat die Kosten für Touch-Interfaces dramatisch sinken lassen – bei weiter steigender Qualität (Auflösung, Größe, Farben, Zuverlässigkeit, Industrie-Tauglichkeit)
- Touch-Interfaces vereinen Anzeige- und Bedienelemente auf engstem Raum und reduzieren Platz- und Kostenaufwand für die Maschinenbedienung
- Touch-Interfaces als Mensch-Maschine-Schnittstelle können zusätzliche Informationen für den Anwender visualisieren (Anwendungshilfen/Handlungsanweisungen, Grafische Darstellung von Störungsursachen, Dash-boards etc.)

White Paper

- Touch-Interfaces sind im Gegensatz zu Tastatur und Maus besser für hygienische Maßnahmen geeignet und werden daher verstärkt u. a. auch in der Medizintechnik eingesetzt.
- Auch umfangreiche Funktionserweiterungen bei Updates können flexibel in Touch-Interfaces aufgenommen werden; die Platzanforderungen der Bedienungselemente bleiben trotz unterschiedlichster Funktionalitäten über Maschinentypen und -generationen gleich

Das volle ökonomische Potenzial von Touch-Interfaces eröffnet sich, wenn das GUI-Design eine intuitive Bedienung von Geräten und Maschinen unterstützt:

- Eine gute, leicht verständliche Bedienerführung, angepasst auf Sprache, Anwendung und Situation, kann die Schulungskosten für das Bedienpersonal erheblich senken
- Verständliche und situationsgerechte Handlungsanweisungen für den Bediener bis hin zu kompletten Workflows für die Konfiguration und/oder die Behebung von Störungen erlauben eine bedienfehlerarme und damit produktivere Anwendung von Geräten und Maschinen
- Über eine verständliche Bedienung lassen sich komplexere Funktionalitäten und spezifische Vorteile der Maschinen für den Anwender nutzbar machen
- Ein modernes, überlegenes Bedienkonzept kann die Produkte eines Herstellers im Wettbewerb differenzieren, die Markenkommunikation verbessern und die Bindung der Kunden intensivieren

b. Warum klassische Entwicklungsverfahren bei Touch-Interfaces versagen

Die klassische Software-Entwicklung verläuft sequenziell. Verkürzt stellt sich die Praxis wie folgt dar: Auf Basis von Spezifikationen wird eine funktionale Software entwickelt. Erst danach – und zumeist von den Entwicklern der funktionalen Software selbst – wird die Bedienoberfläche erstellt. Die führt dazu, dass das GUI meist sehr technik- und funktionsorientiert angelegt wird. Allenfalls nach der Entwicklung des ersten GUIs durch die Software-Entwickler werden Grafiker und Designer hinzugezogen, die dann kosmetische Korrekturen an der Bedienoberfläche vornehmen können. Aus Mangel an Verbindung zu Anwendern und ohne Anwendungswissen können diese jedoch nur das visuelle Design (Icons, Farben und Schriften), nicht aber das Bedienkonzept verändern. Eine Einbindung von Anwendern

White Paper

in den Entwicklungsprozess findet oft nicht oder nur sehr spät statt. Da größere Änderungen an der Oberfläche meist komplexe Änderungen an der Software selbst verursachen, werden Änderungswünsche zu diesem späten Entwicklungszeitpunkt meist mit Hinweis auf die entstehenden Kosten- und Zeitaufwände oder auf den bereits kurz bevorstehenden Launch des Produktes abgelehnt.

Die Folgen dieser Arbeitsweise lassen sich aus den Reaktionen der Anwender ablesen: Bedienoberflächen werden als kompliziert, verwirrend bis hin zu „unbrauchbar“ bewertet. Das aus Perspektive der Softwareentwickler generierte Bedienkonzept entspricht in vielen Punkten nicht den Anforderungen und Bedienabfolgen in der Praxis. Gegengesteuert wird mit teuren Schulungsmaßnahmen. Kurz: Statt ein Gerät intuitiv bedienbar zu machen, wird die Bedienung antrainiert und erlernt. So lässt sich das Produkt später vom gut trainierten Stammpersonal und in Standardfunktionen zuverlässig bedienen – Personalwechsel, der Einsatz wenig genutzter Funktionalitäten oder erweiterter Einstellungsmöglichkeiten führen jedoch zu Fehlbedienungen, sinkender Produktivität und hohen Kosten.

Mittelbar führen die Unsicherheiten und Trainingsaufwendungen auf Kunden- und Anwenderseite sogar dazu, dass die Einführung neuer Maschinenmodelle als extrem kompliziert angesehen und in der Praxis hinausgezögert wird. Spätestens diese indirekten Folgen sollten den Hersteller aus ökonomischem Interesse zum Nachdenken über eine Optimierung seiner Mensch-Maschine-Schnittstelle anregen.

Hinzu kommt: Mit der Verbreitung attraktiver Touch-Display-Benutzeroberflächen in der Consumer-Elektronik entsteht eine unbewusste Erwartungshaltung auf Seiten der Anwender. Der Anspruch an Funktionalität, Ergonomie und ein ansprechendes „Look-and-Feel“ steigt und macht die Entwicklung von HMIs immer anspruchsvoller.

Fakt ist: Die Hersteller industrieller Anwendungen verfügen oft noch nicht über die spezifische Expertise, die nötig ist, um ein intuitives, ansprechendes Touch-Interface-Bedienkonzept zu realisieren. Die folgenden drei Aspekte erscheinen dabei zentral:

- Die Gestaltung der Oberflächen für Touch-Interfaces unterliegt eigenen Regeln. Eine Übertragung aus Usability-Konzepten von PC-Software ist nicht möglich – auch nicht beim Einsatz großer Touch-Displays. Die häufig versuchte 1:1 Abbildung scheitert regelmäßig. Die Einbindung von Usability-Experten, speziellen Tools und neuen Entwicklungsverfahren ist zwingend erforderlich.

White Paper

- Geräteanwender und Maschinenführer benötigen ein schlüssiges, hoch-effizientes Bedienkonzept, das der Anwendungspraxis folgt. Insbesondere die Anwender müssen in den Entwicklungsprozess einbezogen werden.
- Das Design innovativer Touch-Interfaces erfordert eine intensive interdisziplinäre, oft auch standortübergreifende Zusammenarbeit von Applikations-spezialisten, Ergonomen, Entwicklungsingenieuren/Informatikern und Anwendern im Rahmen eines iterativen, nutzerzentrierten Prozesses. Um diese Zusammenarbeit effizient gestalten und die Entwicklungszeiten trotz der Komplexität des Projektes weiter reduzieren zu können, sind neue Methoden und Systeme für die Teamarbeit zu implementieren (Collaboration Infrastructures).

Doch wie genau lassen sich diese Voraussetzungen für eine erfolgreiche Entwicklung innovativer Touch-Interfaces schaffen? Gibt es Best Practice-Ansätze, bewährte Verfahren und Tools, die Entwicklungsleiter und Produktmanager nutzen können, um die eigenen Prozesse und Mitarbeiter in der Entwicklung für die neuen Herausforderungen zu wappnen?

c. Exkurs:

Partnerschaft für innovative Touch-Interface-Softwarelösungen

Das nachfolgend beschriebene Vorgehensmodell wurde aus der Praxis heraus entwickelt. Hauptbeteiligte sind die Softwareentwickler und Software-Infrastruktur-Experten der Berner & Mattner Systemtechnik GmbH, München, und die Usability- und Design-Fachleute der Centigrade GmbH, Saarbrücken.

Beide Unternehmen sind als Dienstleister in zahlreiche Projekte für industrielle Software- und Interface-Entwicklungen eingebunden. Zunächst unabhängig voneinander wurden die Experten beider Unternehmen mit den oben beschriebenen Schwachpunkten klassischer Entwicklungsverfahren konfrontiert und suchten Antworten auf Fragen wie:

- Wie bindet man die Produktexperten des Kunden und die Endnutzer der Produkte so früh wie möglich in die Entwicklung des Interaktionskonzepts für ein Interface ein?
- Wie kann erreicht werden, dass Anwendern möglichst viele Zwischenschritte in der Entwicklung präsentiert werden können und deren Feedback in die laufende Entwicklung von Software und HMI einfließt?

White Paper

- Wie lassen sich in Projekten zu Touch-Interfaces Entwicklungszyklen verkürzen und Fehlentwicklungen vermeiden?
- Lässt sich eine vernetzte Entwicklung beschleunigen und dabei vermeiden, dass Beteiligte aufeinander warten müssen?
- Wie kann die Kommunikation zwischen den Projektbeteiligten, z. B. zwischen Anwendungsentwickler und GUI-Designer oder Kunden und Entwicklern, optimiert werden?
- Wie lässt sich ein Projekt für Änderungen und Varianten offen halten?

Beide Unternehmen erkannten aus ihrer täglichen Arbeit die Notwendigkeit für neue Ansätze und Vorgehensmodelle und etablierten aus dieser Motivation heraus eine Partnerschaft für innovative Touch-Interface-Softwarelösungen. Seither haben die Berner & Mattner Systemtechnik GmbH und die Centigrade GmbH auf Basis ihrer unterschiedlichen Kompetenzen ein Vorgehensmodell entwickelt, das sich auf vorhandene und weiterentwickelte Tools und Prozesse stützt ist. Grundlage bieten dabei bereits etablierte Verfahren wie iterativ-inkrementelle Entwicklungsprozesse, agile Entwicklungsmodelle, der User-Centered Design Process (UCD), Rapid Prototyping und Usability-Tests.

Die ersten gemeinsam entwickelten Produkte sind zwischenzeitlich auf dem Markt. Die Auszeichnungen für diese Entwicklungen, das Echo von Kunden und Anwendern sowie der kommerzielle Erfolg der Produkte können als Beleg für die Eignung des nachfolgend beschriebenen Vorgehensmodells gewertet werden.

3. Vorgehensmodell für die erfolgreiche Entwicklung moderner Touch-Interfaces in industriellen Anwendungen

a. Entwicklung eines Interaktionskonzeptes (Workshop) und Early Prototyping

Damit ein grafisches, interaktives Touch-Bedienfeld am Ende optimal den Erfahrungen und Bedürfnissen des Benutzers in der täglichen Praxis entspricht, müssen die Anforderungen an Funktionen und Darstellung frühzeitig erarbeitet und durch eine iterativ-inkrementelle Weiterentwicklung des HMI fortlaufend überprüft werden.

Hierzu empfiehlt sich folgendes Vorgehen:

Am Anfang der Entwicklung steht ein gemeinsamer „Workshop“ mit allen Stakeholdern und Vertretern von Kunden und Anwendern. Ziel ist die Definition der Anwendungsfälle (Use Cases), um auf dieser Grundlage ein Workflow-orientiertes Interaktionskonzept zu erarbeiten. In der Praxis hat es sich bewährt, die wesentlichen Elemente mit Papier und Bleistift und/oder am Flipchart zu entwerfen und die erforderlichen Interaktionsmechanismen mit allen Beteiligten zu diskutieren. Die Zusammensetzung der Teilnehmer gewährleistet, dass alle Aspekte berücksichtigt werden.

Aus den gemeinsam erarbeiteten Skizzen lassen sich sowohl die grundsätzlichen Interaktionsschnittstellen als auch das Grundlayout der Anwendung ableiten. Erfahrungsgemäß können in entsprechend moderierten Workshops die wichtigsten Funktionalitäten und Bedienelemente definiert werden.

White Paper

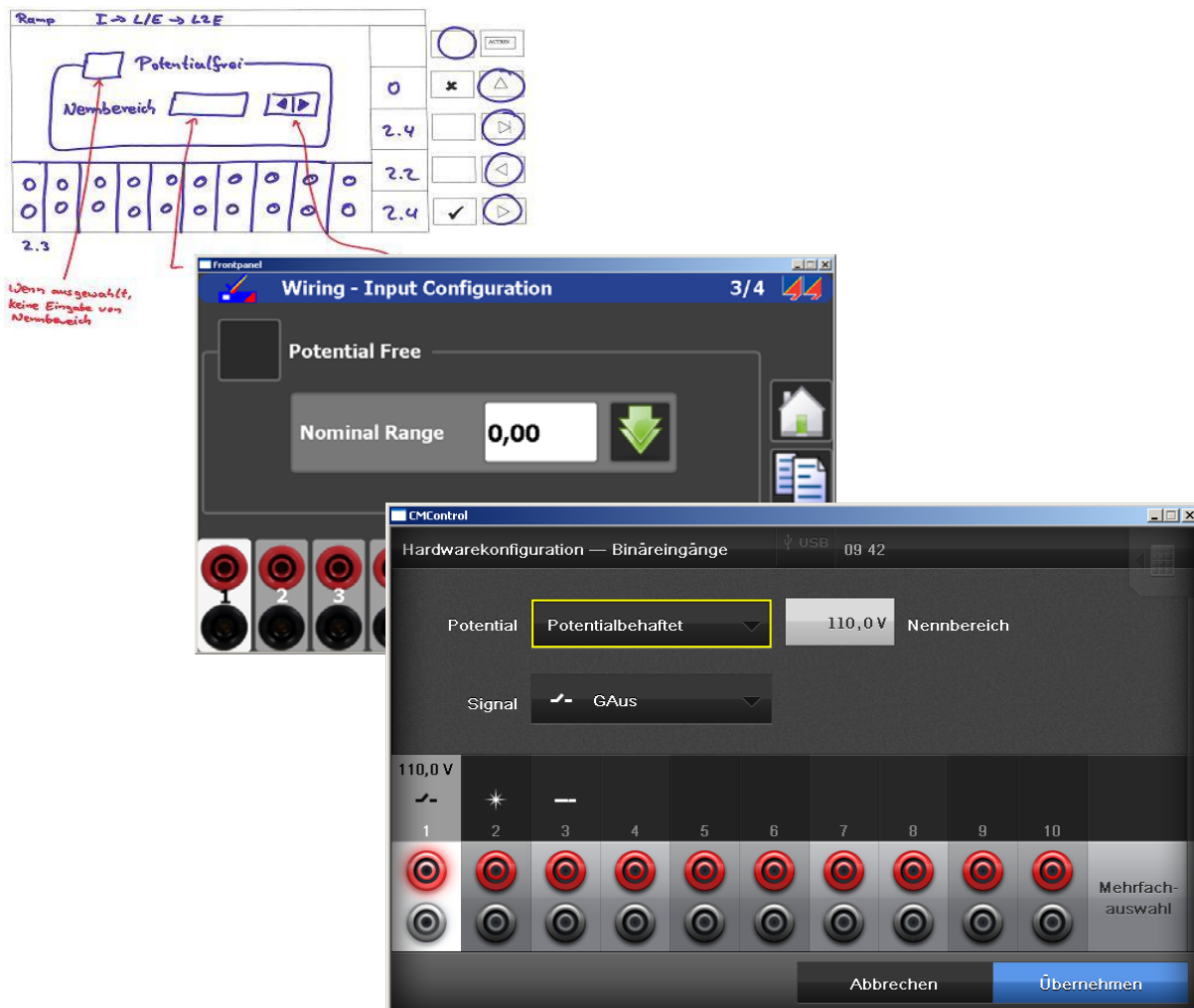


Abb. 1:
Beispiele aus dem Entwicklungsprozess eines Touch-Interfaces für ein Prüfgerät von OMICRON electronics - die ursprüngliche Skizze, ein Prototyp und die Produktversion der Oberfläche.

Auch im weiteren Verlauf des Projektes bleiben die Teilnehmergruppen des Workshops eingebunden. So werden den Anwendern immer wieder Zwischenstände der Touch-Interface-Entwicklung präsentiert. Eingebaut in Mock-ups (funktionsreduzierte Modelle) lassen sich so mit interaktiven Abfolgen von Screen-Designs der praktische Einsatz simulieren und die Bedienbarkeit optimieren – lange bevor ein funktionsfähiger Prototyp des Gerätes verfügbar ist. Die Entwicklung verläuft iterativ und inkrementell, das Feedback der Anwender wird zum frühestmöglichen Zeitpunkt berücksichtigt, was Kosten und Aufwand für Änderungen minimiert.

b. Parallele Entwicklung von funktionaler Software und GUI-Design

Die zweite Komponente des Vorgehensmodells ist entscheidend für die Effizienz im Entwicklungsprojekt. Durch die inkrementelle Entwicklung des GUI unter aktiver Beteiligung der Anwender verändern sich die Prioritäten bei der Interface-Entwicklung. Anders als bei den klassischen technisch-funktionalen Designs setzt der Anwender die Prioritäten: „Der Schwerpunkt liegt auf dem Bedienablauf, nicht auf der Funktionsimplementierung.“

Aber leidet der Software-Entwicklungsprozess dann nicht unter der schrittweisen Optimierung der Bedienoberflächen im Dialog mit dem Anwender? Nein, denn der konsequente Einsatz von neuen Software-Entwicklungswerkzeugen wie den GUI-Beschreibungssprachen XAML (Microsoft, .NET), MycroXAML (OpenSource, C++), SwiXML (OpenSource, Java) und QML (Nokia Qt, C++) ermöglicht es, das Design der Oberflächen annähernd komplett von der Entwicklung der funktionalen Software zu entkoppeln. Beide Entwicklungsprozesse können somit parallel abgearbeitet werden.

Während die GUI-Designer und Usability-Experten in enger Abstimmung mit den Anwendern die Bedienoberfläche auf Basis des Interaktionsmodells verfeinern und dafür ihre speziellen Tools wie Adobe Photoshop zur Erstellung von Pixel-präzisen Screen Designs sowie MS Expression Blend zur Erstellung von dynamischen und interaktiven Prototypen nutzen, arbeiten die Software-Entwickler an den Funktionalitäten und der Interaktion mit der Hardware – auf Basis des gemeinsam entwickelten Interaktionskonzeptes. Mit Hilfe der Seitenbeschreibungssprachen – und damit in einer Datei parametrisierbar – erfolgt die Verbindung von Funktion und Oberflächendesign. War es in der klassischen Softwareentwicklung bisher aufwändig, Funktionalitäten von einer speziellen Oberfläche auf eine andere zu verlegen, so kann dies jetzt über die Einstellung via Parameter auf Wunsch des Anwenders vom Designer erledigt werden – der dahinterliegende Funktionsaufruf in der Software wandert automatisch mit.

Mit GUI-Beschreibungssprachen haben Designer und Programmierer erstmals eine gemeinsame Basis und können Funktion und Layout parallel entwickeln, ohne sich gegenseitig zu behindern. Professionelle WYSIWYG-Editoren für diese Sprachen geben den Designern die Freiheit der Gestaltung und der Anreicherung mit interaktiven Abläufen, ohne dass diese sich um den Code kümmern müssen. Daraus ergibt sich auch ihr großer Vorteil für die schnelle und nach Design und Funktion getrennte HMI-Entwicklung: Die Bedienoberflächen können ohne Softwareänderung flexibel konfiguriert werden. Die Realisierung von Produktvarianten, z. B. für

länderspezifische oder kundenspezifische Versionen, ist damit sehr einfach umzusetzen. Dies gilt auch für den Einsatz auf verschiedenen Betriebssystemplattformen.

c. Infrastruktur für eine vernetzte, agile Entwicklung (Distributed Development Platform)

Das hier beschriebene Vorgehensmodell mit seinen inkrementell-interaktiven Entwicklungsschritten und der engen Beteiligung von Anwendern und Designern erfordert eine sehr intensive Kommunikation innerhalb des - gegenüber klassischen Entwicklungsverfahren vergrößerten - Projektteams. Die teilweise parallelisierten Entwicklungsstränge können ohne ständigen Informationsaustausch, Zugriff auf die jeweils aktuellen Software-Versionen und vollkommene Transparenz des Projektfortschritts für alle Beteiligten nicht zum gemeinsamen Erfolg geführt werden.

Mit gutem Willen allein ist diese Kommunikation nicht zu etablieren, zumal durch die Beteiligung von Anwendern, Hardware- und Software-Spezialisten und externen Dienstleistern zumeist auch überregional oder sogar international kooperiert werden muss. Zuverlässig lassen sich solche Entwicklungsprojekte nur über den Einsatz einer entsprechenden Software-Infrastruktur zum Erfolg führen.

Die von Berner & Mattner entwickelte **Distributed Development Platform** ist eine umfassende Entwicklungsplattform, die weltweit mit definierten Zugriffsrechten benutzt werden kann und auf das jeweilige Projekt angepasst wird. Diese Infrastruktur unterstützt das Projektteam über folgende Funktionen:

- Rollenbasierte Zugriffsrechte
- Ticketing-System für ein lückenlos rückverfolgbares Aufgabenmanagement; insbesondere können Änderungswünsche und ihre Verbindung zu den Code-Versionen zuverlässig erfasst, kommuniziert und deren Abarbeitung kontrolliert werden
- Projektmanagement-Funktionalitäten (Projektplan, Zeiten, Milestones, Fortschritte etc.)
- Sourcecode-Verwaltung mit Konfigurationsmanagement für die Versionen inklusive deren Historie
- Build-Verwaltung und automatische Generierung der Artefakte für den Zugriff aller Projektteilnehmer auf den jeweils aktuellen Softwarestand
- Einbindung automatisierter Regressionstests insbesondere für HMI-Software (Varianten, Mehrsprachigkeit, Testfälle für typische Interaktionsfolgen)

White Paper

Die Plattform ermöglicht kurze Zyklen und damit eine moderne, agile Softwareentwicklung. Auf Knopfdruck oder ausgelöst durch das Einstellen einer Änderung wird der HMI-Buildprozess automatisiert durchgeführt; den verschiedenen Entwicklungsteams steht damit immer der aktuelle Softwarestand zur Verfügung. Die Einbindung von Kunden und Anwendern ist zu jedem Zeitpunkt möglich. Der Entwicklungsprozess erreicht ein extrem hohes Maß an Transparenz.

Die Möglichkeit, automatische Regressionstests direkt in diese Infrastruktur einzubinden, ist insbesondere für agile HMI-Software-Entwicklung entscheidend, um die zunehmende Funktionalität in der Entwicklung kontinuierlich absichern zu können. Tools wie GoogleTest bieten für HMIs die Möglichkeit, einen virtuellen Nutzer zu simulieren. Dabei werden Benutzeraktionen aufgezeichnet und als automatische Testfälle wiederverwendet. Große Teile von Software und HMI können so effizient und vollständig automatisiert getestet werden. Für Qt-basierte GUI-Applikationen ist damit auch eine automatische Textlängenverifikation für alle GUI-Elemente und somit die Überprüfung aller Sprachversionen auf Knopfdruck möglich.



Abb. 2:
CMControl Bedienterminal für das CMC-Prüfgerät von OMICRON electronics
– mit dem iF communication design award 2010 ausgezeichnetes Design von
Centigrade

4. Fazit

Hersteller und Projektverantwortliche, die ihre industriellen Produkte über intuitiv bedienbare Touch-Interfaces und herausragende GUI-Designs differenzieren wollen, verfügen mit dem oben beschriebenen Vorgehensmodell über einen Best Practice Ansatz. Dieser adressiert über seine Tool-Architektur die wesentlichen Anforderungen an einen effizienten, parallelisierten Entwicklungsprozess, schafft mit einem Interaktionskonzept die Basis für ein teamübergreifendes Verständnis der praktischen Anwendung und garantiert die frühe und fortlaufende Einbindung von Anwendern und Usability-Experten.

Die Erfahrung zeigt, dass die Projektteams der Hersteller neue Arbeitsweisen und neue Tools zum Einsatz bringen müssen. Die frühzeitige, intensive Einbindung von Anwendern sowie Design- und Usability-Experten erfordert ein neues Denken und Methoden, die Elemente der agilen Software-Entwicklung nutzen. Die damit einhergehende intensive Abstimmung zwischen allen Projektbeteiligten erzwingt den Einsatz effektiver Werkzeuge zur verteilten Entwicklung – auch, um gleichzeitig die notwendige Effizienz und Transparenz in den Gesamtprozess zu bekommen.

Derart grundlegende Veränderungen von Tools und Arbeitsweisen erfordern von den Herstellern ein professionelles Management dieses Änderungsprozesses, um die eigenen Mitarbeiter nicht zu überfordern. Da die meisten Entwicklungsprojekte unter hohem Zeit- und Erfolgsdruck stehen, sollten Hersteller bei den ersten Projekten zur Touch-Interface-Entwicklung erfahrene externe Spezialisten und aufgeschlossene eigene Mitarbeiter in einem Team zusammenzuführen. So lassen sich die neuen Arbeitsweisen leichter erlernen und erste Praxiserfahrungen mit diesen Methoden und Tools sammeln.

Kontakt

Anregungen oder weitergehenden Fragen zu diesem White Paper sowie zur Entwicklung innovativer Touch-Interfaces allgemein können sie richten an:

Dr. Klaus Wiltschi, Berner & Mattner Systemtechnik GmbH
klaus.wiltschi@berner-mattner.com, +49 (0)89 608090-0

Dr. Michael Sturm, Berner & Mattner Systemtechnik GmbH
michael.sturm@berner-mattner.com, +49 (0)89 608090-0

Thomas Immich, Centigrade GmbH
thomas.immich@centigrade.de, +49 (0)681 959 311 0

Firmenprofile

Berner & Mattner Systemtechnik GmbH, München

Das Unternehmen Berner & Mattner ist Spezialist für die Spezifikation, Entwicklung und den Test komplexer Elektroniksysteme. Sein branchenübergreifendes Leistungsspektrum reicht von der Beratung, Konzeption, Software- und Systementwicklung bis hin zum Aufbau und Betrieb kompletter Test- und Integrationssysteme. Für die Branchen Automobil, Verteidigung, Luft- und Raumfahrt, Schienenverkehr und Industrieautomation liefert Berner & Mattner maßgeschneiderte Software- und Engineering-Lösungen auf Basis seiner Produkte und Dienstleistungen. Mit seinem durchgängig modellbasierten Ansatz optimiert Berner & Mattner die Effizienz und Qualität in der Software- und Systementwicklung seiner Kunden. Daher vertrauen namhafte Unternehmen wie AUDI, BMW, Daimler, Deutsche Bahn, EADS Deutschland, Siemens u.v.m. der Kompetenz von Berner & Mattner. Das 1979 gegründete Unternehmen mit Hauptsitz in München beschäftigt derzeit an sieben Standorten in Deutschland und Österreich 300 MitarbeiterInnen und ist Mitglied der internationalen Assystem Group.

Weitere Informationen über Berner & Mattner: www.berner-mattner.com

Centigrade GmbH

Centigrade unterstützt Unternehmen bei der Entwicklung von User Interfaces, die sich durch hohe Benutzerfreundlichkeit, visuelle Attraktivität und technische Eleganz auszeichnen. Dank eines kreativen und agilen Teams von Usability-Experten, Designern und Software-Architekten kann Centigrade eine Reihe maßgeschneiderter User-Experience-Dienstleistungen bieten. Das Spektrum reicht von der Anforderungserhebung über die Usability-Evaluation bestehender Systeme bis hin zur Konzeption und Erstellung exklusiver Screen- und Icon-Designs. Darüber hinaus implementiert Centigrade spezialisierte GUI Controls und Look and Feels für Anwendungen, die auf Java Swing, WPF oder Silverlight basieren. Zu den Kunden von Centigrade gehören sowohl mittelständische Unternehmen als auch international agierende Konzerne verschiedenster Branchen. Die Arbeiten des Unternehmens wurden unter anderem mit dem iF communication design award ausgezeichnet.

Weitere Informationen über Centigrade: www.centigrade.de

Weitere Informationen zum Thema

„Usability Engineering kompakt: Benutzbare Software gezielt entwickeln“
Michael Richter, Markus D. Flückiger
Spektrum Akademischer Verlag; 2. Auflage, Mai 2010

„About Face – The Essentials of Interaction Design“
Alan Cooper, Robert Reimann, David Cronin
Wiley Publishing Inc., 2007 - <http://www.cooper.com/>

„Understanding Mobile Human-Computer Interaction“
Steve Love, Elsevier 2005

„Mobile Usability“
Ch. Lindholm/ T. Keinonen/ H. Kiljander
McGraw-Hill 2003

„Entwicklung benutzerorientierter Embedded MMIs : Den Anwender im Blick“
Alexander Wiethoff, Alexander Sorg
Article in "Markt und Technik" Issue 49/2009, ISSN 0344-8843

„Use Cases – effektiv erstellen“
Alistair Cockburn,
Mitp-Verlag, 2008

User centered Design Process
<http://www.w3.org/WAI/redesign/ucd>

XAML - <http://msdn.microsoft.com/de-de/library/ms752059.aspx>

mycroXAML - <http://www.codeproject.com/KB/dotnet/MycroXaml.aspx>

SwiXML - <http://www.swixml.org/>

Qt - <http://qt.nokia.com/>

QML - <http://doc.qt.nokia.com/4.7-snapshot/qdeclarativeintroduction.html>

Googletest - <http://code.google.com/p/googletest/>

Berner & Mattner Systemtechnik GmbH - www.berner-mattner.com

Centigrade GmbH – www.centigrade.de

OMICRON electronics GmbH – www.omicron.at