



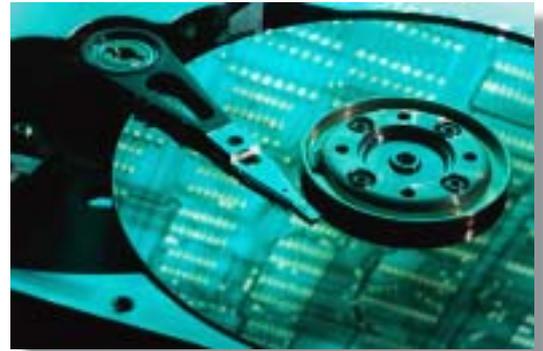
Portierbarer Code

Am Beginn von Entwicklungsprojekten stellt sich oft die Frage, wie weit das zu erzielende Ergebnis portierbar sein soll. Mit anderen Worten, genügt es, wenn die Software hier und jetzt auf genau diesem Rechner lauffähig ist? Oder ist es notwendig, die Übertragung der Software in andere Umgebungen zu einem späteren Zeitpunkt einzuplanen?

Die Gründe für eine allfällige Portierung können vielfältig sein: Zum Beispiel kann ein Hardwarewechsel zur Performancesteigerung gewünscht werden, ein Upgrade oder gar Wechsel des Betriebssystems anstehen oder der Einsatz einer neuen Entwicklungsumgebung fällig sein. Jedenfalls sagt einem allein der gesunde Hausverstand, dass man bei den kurzen Innovationszyklen der IT-Produkte gut daran tut, die Portierbarkeit von Software von vorneherein im Auge zu behalten. Es gibt aber noch einen weiteren Grund, der für die Erstellung von portierbarem Code spricht – die Qualität der erzeugten Software. Wird nämlich auf die Portierbarkeit geachtet, dann setzt das ein sehr sauberes Arbeiten voraus. In weiterer Folge bringt das, dass der Code leichter wartbar ist. Die Erzeugung von portierbarem Code hat also eine Art Katalysatorfunktion für die Qualität der Software.

Die Profis der InfraSoft haben einen umfangreichen Fachbeitrag zusammengestellt, der viele wertvolle Informationen zum Thema „portierbarer Code“ enthält. Neben einer Beschreibung der unterschiedlichen Portierungsfälle gibt der Beitrag praktische Hinweise, was in diesen Fällen zu beachten ist und wie portierbarer Code am besten zu erzeugen ist. Interessierte finden den Text „Portierbarer Code“ auf der InfraSoft Homepage unter www.infrasoft.at/service

zum kostenlosen Download. Mit den folgenden Zeilen möchten wir Ihnen einen ersten Überblick geben, worum sich die Fragestellungen im Thema „Portierbarer Code“ drehen.



Portierbarer Code sichert nicht nur gegen einen möglichen Wechsel der Systemumgebung ab. Auch Qualitätsanforderungen werden dadurch elegant abgedeckt.

Wann ist Code portierbar?

Unter Codeportierung versteht man generell das Adaptieren von Programmcode für den Einsatz in einer anderen Systemumgebung. Unter einer Systemumgebung ist dabei eine bestimmte Kombination von Computer, Betriebssystem und Entwicklungstool zu verstehen. Natürlich ist im Grunde jeder Code portierbar - was jedoch als wesentliche Codeeigenschaft im Zuge einer Portierung auftritt, ist die Schwierigkeit oder eben die Leichtigkeit des Portierungsprozesses. Unter „portierbarem Code“ versteht man daher generell Code, der mit verhältnismäßig geringem Aufwand an eine andere Umgebung angepasst werden kann.

In der Praxis gibt es zumindest drei sehr unterschiedliche Arten von Portierung: die Portierung von Computer zu Computer, die Portierung von einem Betriebssystem auf ein anderes und nicht zuletzt der Wechsel der Entwicklungsumgebung. Bei einer konkreten



Portierungsaufgabe können natürlich auch beliebige Kombinationen dieser Teilaufgaben auftreten:

- *Wechsel der Hardwareplattform:* Ein Wechsel der Hardware wird typischerweise dann notwendig werden, wenn man gestiegene Performanceanforderungen erfüllen bzw. eine erhöhte Last abdecken möchte. Andere Gründe könnten eine erhöhte Ausfallsicherheit der neuen Hardware sein. Aber auch firmenpolitische Gründe, wie etwa Verträge mit Hardwareherstellern, können hier ausschlaggebend sein.
- *Wechsel des Betriebssystems:* Ein Betriebssystemwechsel hat entweder einen Hardwarewechsel zur Ursache, oder das verwendete Betriebssystem wird nicht mehr vom Hersteller unterstützt und muss daher langfristig ausgetauscht werden. Oft, aber nicht immer, kann mit einer Nachfolgeversion vom selben Hersteller gerechnet werden. Die Nachfolgeversion ist meistens soweit wie irgend möglich kompatibel zur Vorgängerversion.
- *Wechsel der Entwicklungsumgebung:* Ein solcher Wechsel wäre etwa der Wechsel des Compilers und damit zumeist auch der Wechsel der mitgelieferten Bibliotheken (sowohl der Runtime Library als auch der Betriebssystemanbindung). Für einen Wechsel dieser Art gibt es wahrscheinlich die meisten guten Gründe und er wird wohl auch am häufigsten passieren. Zum Beispiel könnte der Hersteller des alten Compilers den Support einstellen oder einfach vom Markt verschwinden, der alte Compiler könnte nicht mehr ausreichend gewartet werden, oder ein neuer Compiler könnte schnelleren oder besseren Code produzieren.

InfraSoft als Partner

bietet Ihnen professionelle Unterstützung.

- **Wir setzen modernste Methoden ein**
Unsere Mitarbeiter verfügen über fundierte Kenntnisse in C++, Java, SQL, Notes und HTML/XML.
- **Wir stellen Know-how bereit**
Die InfraSoft-Teams haben langjährige Erfahrungen mit Datenbanken, Networking und Internet.
- **Wir schonen Ihr Budget**
Unsere Entwickler stehen *standby*. Sie bezahlen uns nur dann, wenn Sie uns brauchen.
- **Wir übernehmen Verantwortung**
Unsere Teams sind bereit, komplexe Probleme in Eigenregie zu lösen.

Die genaue Beschreibung dieser drei Portierungsfälle (auf der InfraSoft Website kostenlos zum Download unter www.infrasoft.at/service) wurde von den Profis der InfraSoft erstellt. Die Autoren stehen unter anderem auch für Beratungen zur Verfügung, wie Programmcode möglichst portierbar ausgelegt werden kann. Code Reviews und spezielle Programmier-Trainings durch die InfraSoft-Profis stellen dabei wertvolle ergänzende Dienstleistungen dar. Mehr Informationen erhalten Sie auf Anfrage unter info@infrasoft.at.



Die Profis von InfraSoft sind interessant für mich. Bitte kontaktieren Sie mich:

Vorname : _____
Nachname : _____
Firma : _____
Adresse : _____

Tel : _____
Handy : _____
Fax : _____
e-mail : _____
via http:// _____

