



Modern Packaging for Windows Platform



Microsoft® Partner

Gold Software Development

Gold Web Development

Silver Independent Software Vendor (ISV)

Arcadia, ZAO
30 korp. 2A, Zanevskiy pr.,
St. Petersburg, Russia
P.O.B. 8, 195112
www.offshore-software.ru

Tel: +7 812 610 59 55
+7 812 610 59 58
+358 94 247 3303
Fax: +7 812 610 59 57
info@arcadia.spb.ru



Packaging

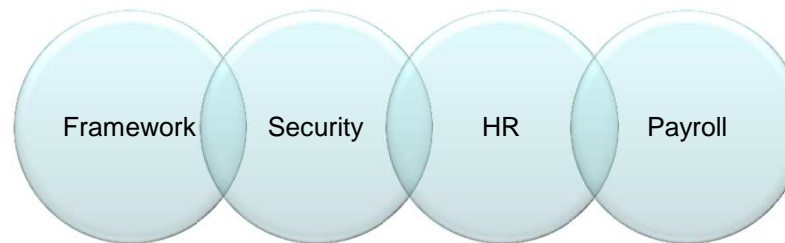
...is the science, art, and technology of enclosing or protecting products for distribution, storage, sale, and use...

From Wikipedia, the free encyclopedia

Background



- Previously with consumer electronics software and hardware
- Started as a QA in Arcadia, worked in different roles and nearly all projects thru out the customer;
- Now hold Team Lead position in the Software Packaging department
- Big Scandinavian customer, HR and Financial software solutions
- About 10 retail and some internal / framework projects



Background

- Heterogenic platforms, integration and standardization issues, now Microsoft technology oriented
- Strict requirements and usually time critical deliveries
- Multiple subcontractors,
- Work with international, distributed Scrum teams
- Frequent personal communication
- **Significantly improved customer feedback**



Globally – Windows Packaging

- This area is poorly standardized in general, except Windows Installer
- No significant research or attempts to provide complex patterns
- Requires system hardware and deep software configuration knowledge
- Big amount of custom vendor specific solutions different by design
- Most organizations have their own internal installation frameworks
- **Quality is critical** - error localization is complex and resourceful



Purpose



- To share the experience
- To get the feedback
- To know more about latest news in this area

Welcome to the REAL WORLD



Typical Errors



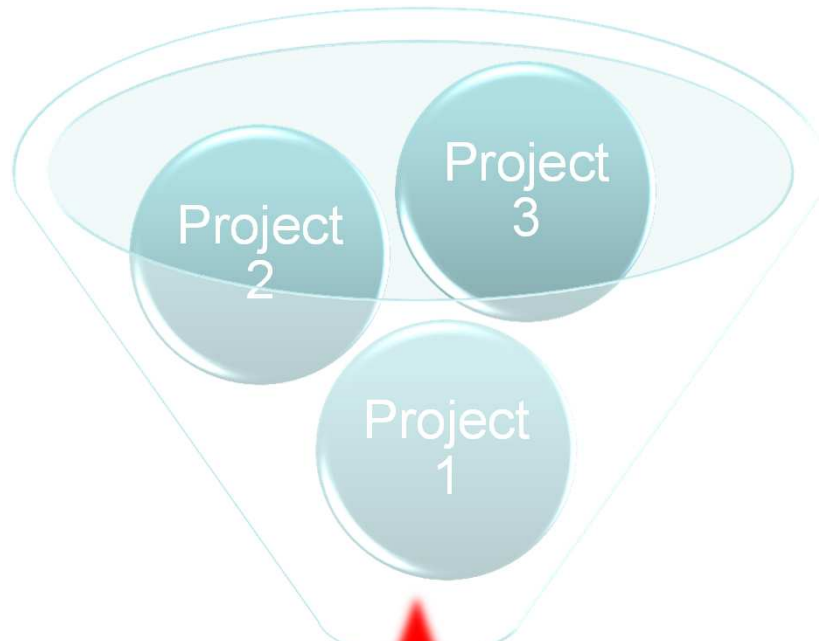
- Develop installation as a software product with the same or lower quality
- Change internal package design without informing customer
- Prioritize application features and quality in expense of packaging
- Start packaging soon before the critical delivery

Typical Errors



- Don't invest in the packaging design and infrastructure
- Remove garbage from the build outputs right before the final delivery
- Try to automate process without knowledge of the customer requirements
- Limited ability to say "No" and in decision making
- Try to integrate in every project ecosystem completely

Packaging - Inputs



Conflict with Scrum!

Packaging Team

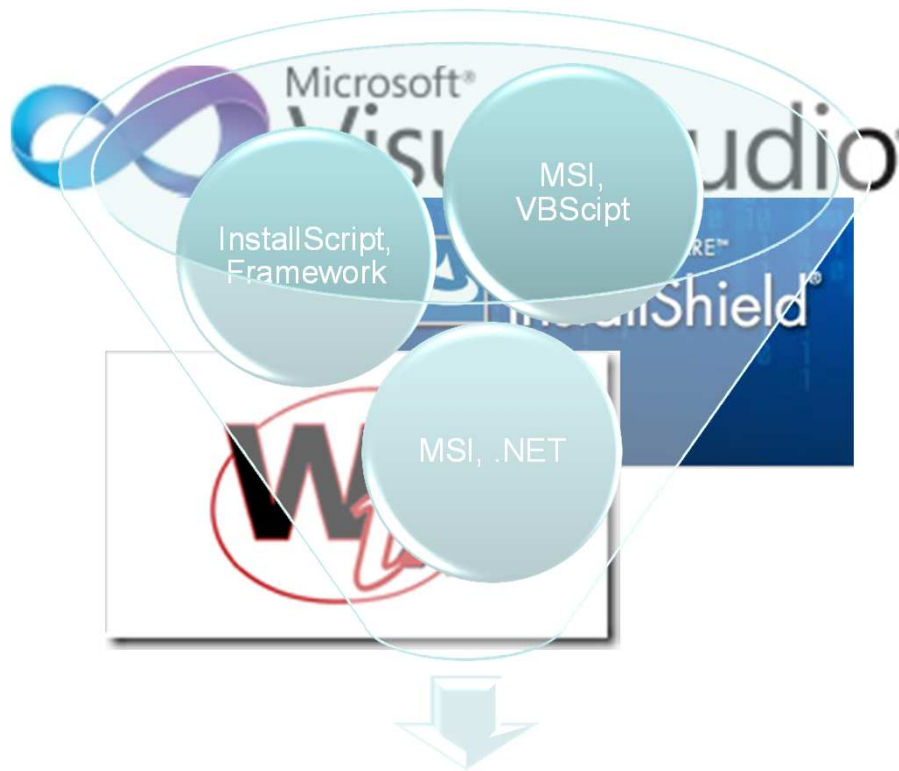


Packaging – Outputs



1. Gather the application requirements
2. Get the initial deliverables version
3. Provide feedback and ensure that you can satisfy most stakeholders
4. Agree on the common rules and achieve acceptable compromise if needed
5. How to localize application related errors
6. Create acceptance and internal test cases
7. Create test environment if possible
8. Create or update installation package
9. Save sources and binaries
10. Delivery for the internal, application and customer care testing
11. Documentation update

Development Platforms



- Source Safe, Team Foundation Server 2009, now 2010
- Windows Installer in general
- InstallShield 11-2011, VBScript, InstallScript, now mostly WIX DTF
- Visual Studio 2008, 2010, .NET 2.0 – 4.0
- Highly fragmented proprietary installation framework as a basis used by consultants based on InstallShield/VBScript

Windows Installer XML



Microsoft®

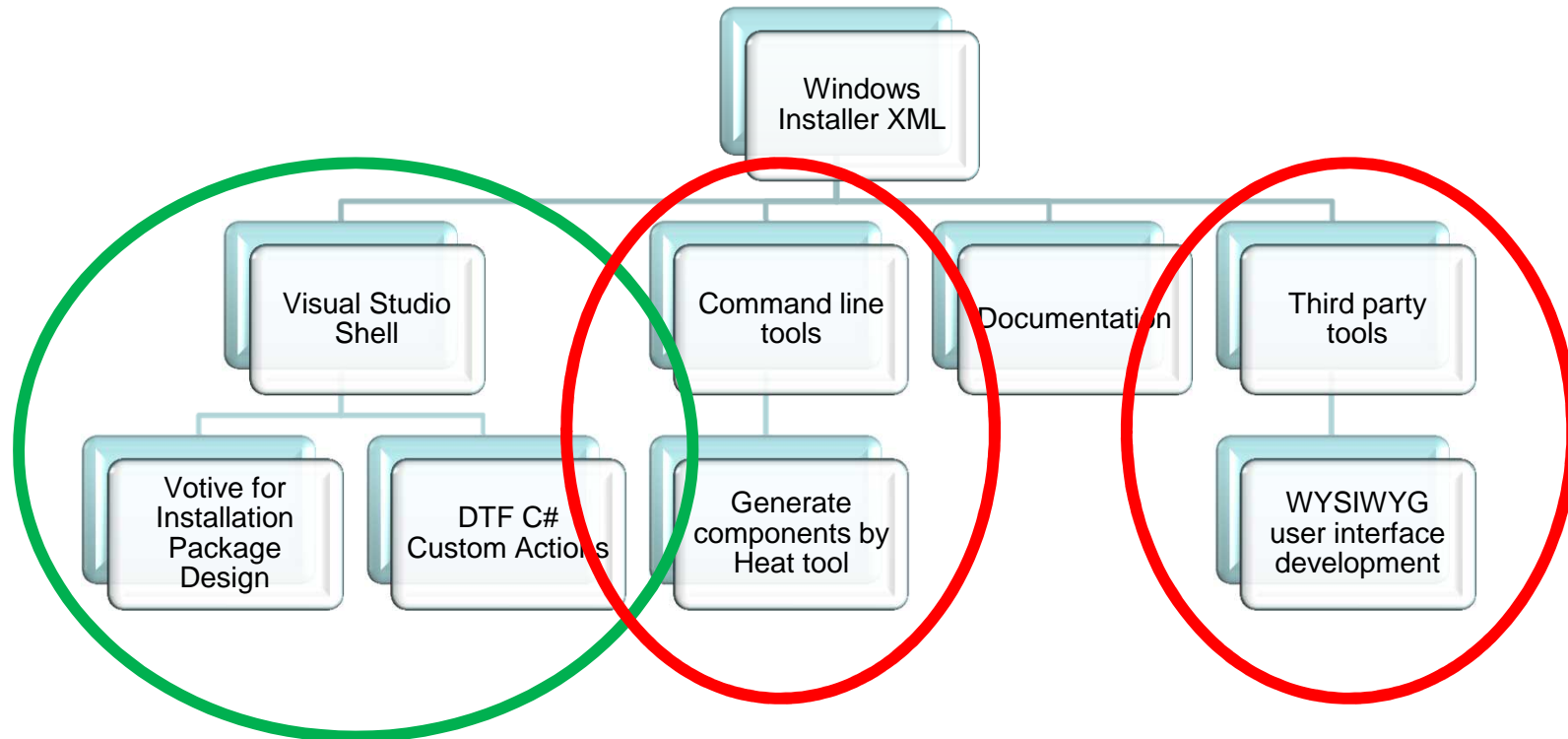
- WiX was developed by Microsoft software design engineer Rob Mensching
- Released to SourceForge in April of 2004 – first Microsoft open-source project
- Developers use their free time to participate in the development
- Everyone can join the project – code changes submitted via maillist
- After some successful changes – direct access could be provided

InstallShield

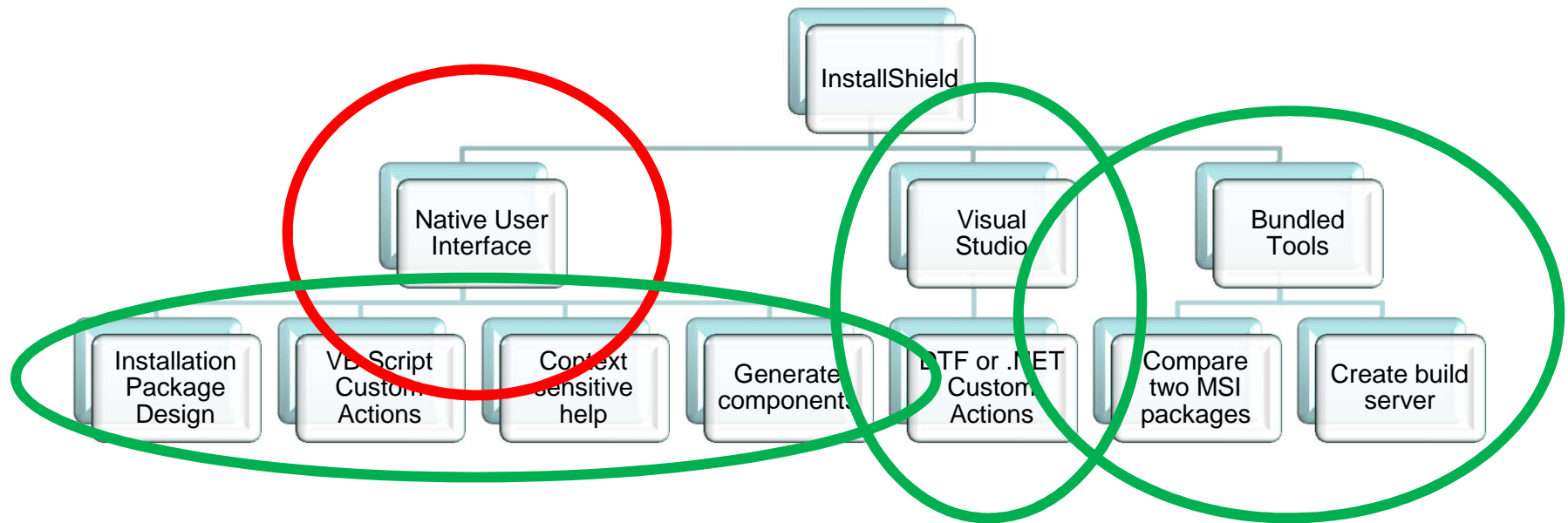


- Since 1992 InstallShield was developed by Stirling Technologies
- Now InstallShield developed by Flexera Software owned by Thoma Cressey Bravo private fund
- Known industry standard
- Formerly InstallScript solution
- Successfully implemented support for Windows Installer technology
- Widely used by our Customer in the most of the projects
- Now Limited Edition shipped with Visual Studio 2010

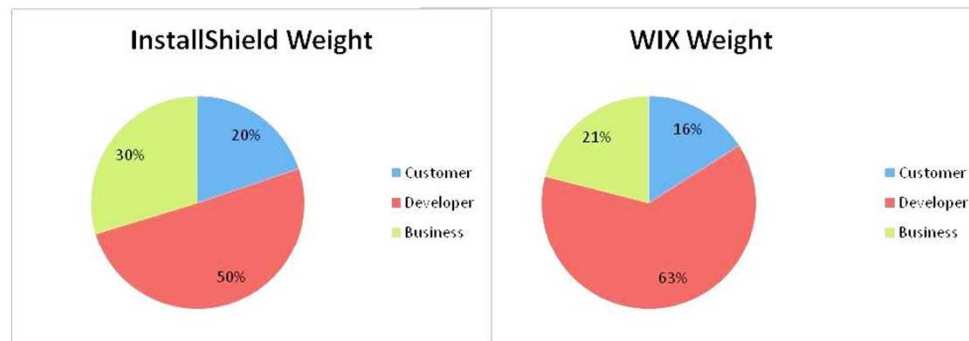
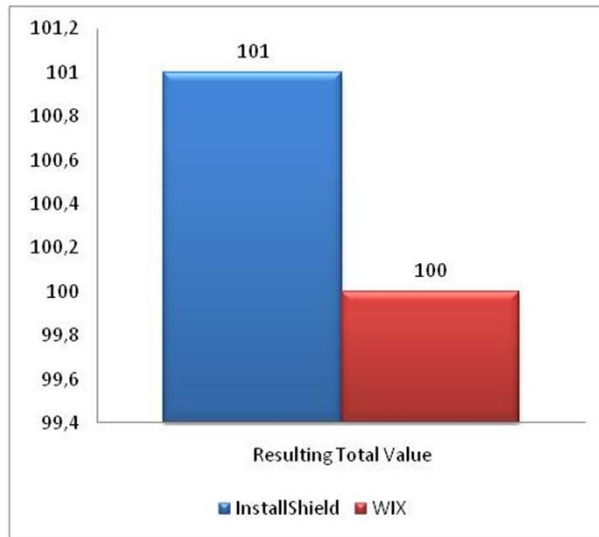
WIX – Current Use Case



InstallShield – Current Use Case



WIX vs InstallShield – Summary



● InstallShield

- ▶ Globally InstallShield still standard solution for the market
- ▶ More predictable, supported and controllable product
- ▶ Development is not too complicated, best installation visualization
- ▶ Very expensive

● Windows Installer XML

- ▶ Some concerns in regard of the customer orientation
- ▶ Business value is questionable, since the product life cycles and features hard to control or predict
- ▶ Best solution from developers point of view and a key of success
- ▶ Free

As the result, compromise...



InstallShield 2011 and WIX Deployment Tools Foundation

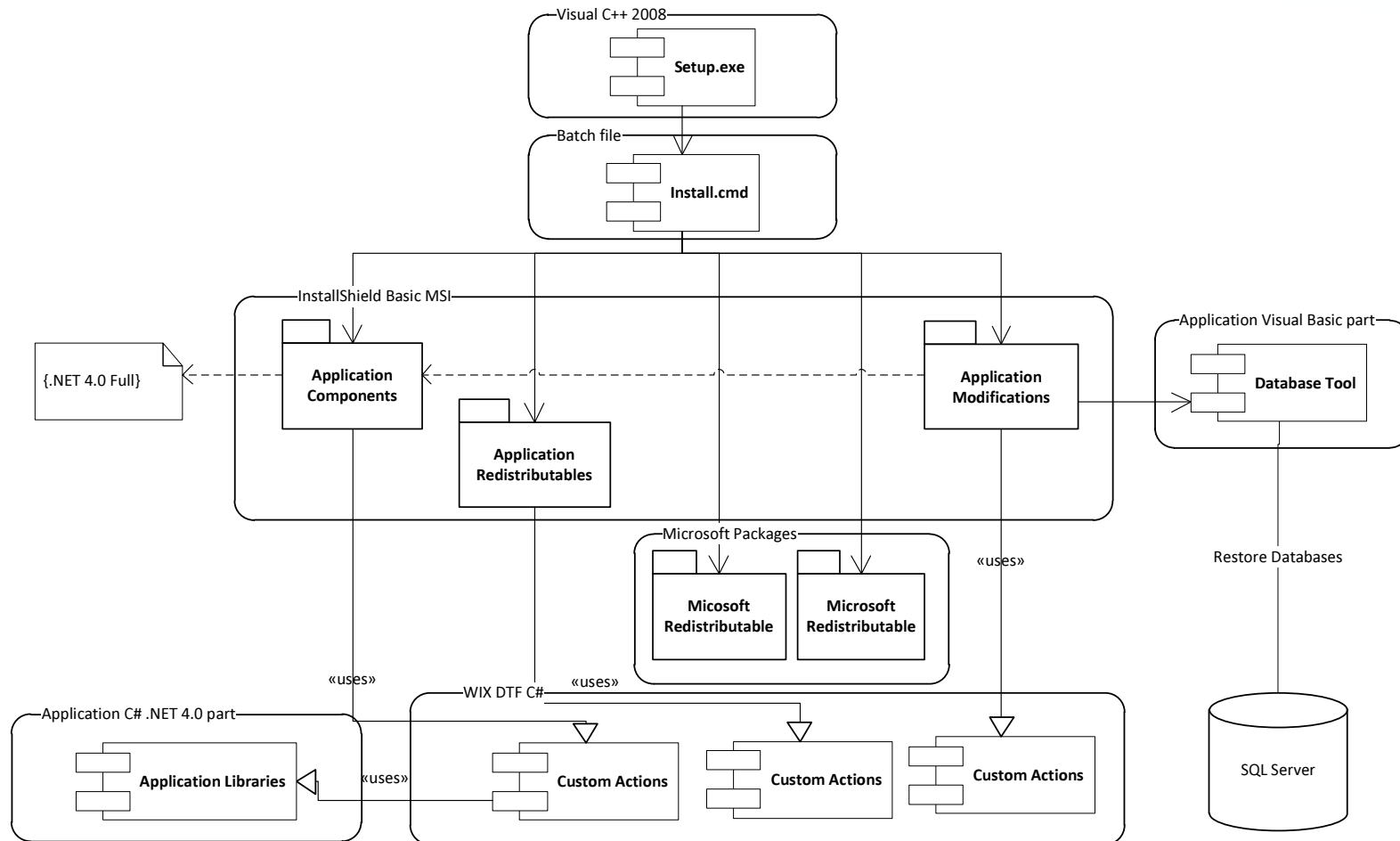
- Cheaper than Premier Edition
- Concurrent licenses
- Multilingual support could be made in external UI
- InstallShield UI allows fast visualization and editing of the MSI package
- WIX DTF and C# code handles most of the custom actions needed, especially for Silverlight and .NET based ones
- Now InstallShield bundled with the Visual Studio

Automation Pros & Cons



- When automate
 - ▶ Stable and mature project
 - ▶ Well documented
 - ▶ Installation functional, delivered and approved by the customers
 - ▶ Frequent updates
 - ▶ Rare customer specific releases
- Where it is not needed
 - ▶ Requirements are not even or stable
 - ▶ Frequent and significant application design changes
 - ▶ Rare updates, slow development
 - ▶ Highly variable customer demands
 - ▶ Customer specific deliveries
 - ▶ Process is not documented
 - ▶ Limited resources or budget

Experimental Design Pattern

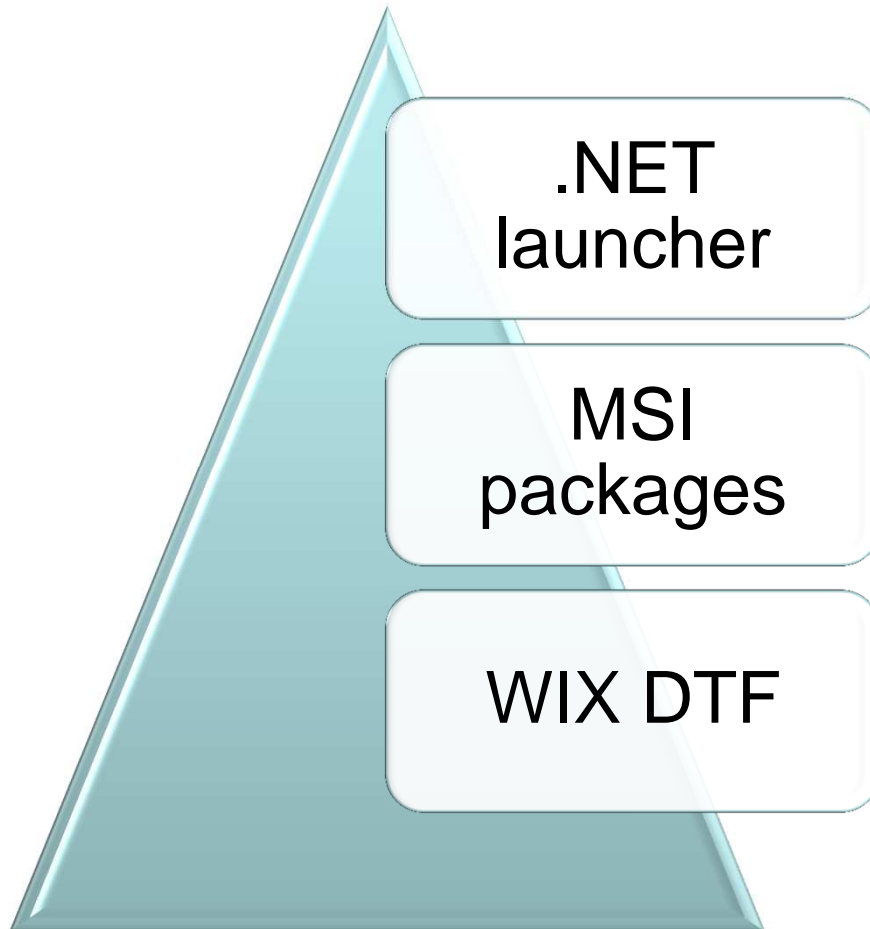


Experimental Design Pattern



- Two packages per product
 - ▶ Components for the files
 - ▶ Modifications for the environment configuration
- Easy to localize the issue
- Faster to solve configuration related errors
- Reusable assemblies from the application
- Functional template based on the InstallShield and WIX DTF, ready for the build automation
- Automated documentation processing

Future of Packaging



- InstallScript out of the market
- Windows Installer user interface will be hidden by custom .NET based solutions
- Silent mode used more and more
- Multi instance design as the requirement for growing ASP markets
- No VB or InstallScript inside the installation - .NET, C# and WIX DTF based solution
- Integration with the application assemblies
- Flexible build automation that will handle most of the customer related cases
- ...APPX and Windows Marketplace?...

...So, Don't be Just a Setup Monkey



Q&A