Disclosures

- The Department of Radiation Oncology at the University of Alabama has several research agreements with Varian Medical Systems.
- Research supported in part by Varian Medical Systems
- Teaching honoraria received Varian Medical Systems

Eclipse™ Scripting API (ESAPI) Overview

- Programming Interface to ARIA™/Varian Apps
- Abstraction with some hand holding
- Eclipse Plugin
- Standalone

Aria Database / Varian Application
The Abstraction

- Table structure to intuitive objects

.NET Framework

- Supported Languages: C#, F#, VB.net, Iron Python
- Collection of common code libraries installed on Windows
- Intellisense support

WPF – Windows Presentation Foundation

- Rich User Interfaces
- 3D graphics
- Same framework used in Varian native apps
- Also supports Console applications, and Windows Forms
What Can I Do With It?

- Can currently only read from database
- Plan Quality Checks
  - DVH Analysis – Does this meet our standards?
  - Structures named appropriately
- Custom tools / Plugins
- Easy data mining
  - Access CT pixels, dose voxels, structure contour 3D positions, control points, MLC

Datamining Work

- Integration of functional rest in radiation treatment planning system
- Journal of Clinical Dentistry

Example XCheck

- Runs as plugin to Eclipse
- Used by dosimetrist
- Standards in planning DVH by site
- User Interface
- Reports generation
The Future of ESAPI

- Clinical writing!
- Field arrangement
- Plan generation/Optimization
- Structure creation
- New Tools for Easier Code Generation

Getting Started

- Basic Tools
  - Varian Eclipse 11.0 +
    - No additional cost
  - Visual Studio Community Edition
    - Free
  - Imagination

Where To Learn More

- Varian Developer
- Codeplex - strong community
- Oncopeer.com
- Developer workshop
- Youtube ("ESAPI")/CarlosAnderson.com
- Explore with Visual Studio and Intellisense
- .NET Programming
  - Pluralsight
  - Microsoft Virtual Academy
Guidelines?

• Current Risks
  • Clinical decision making
  • Bad data collection for research

• Future Risks
  • Incorrect structure creation
  • Non-optimal planning

Three Tiered Approach to In-House Clinical Software Rollout

1. Unit testing critical software pieces
   • Minor/Major code errors
2. Beta launch with small subgroup of advanced users
   • Usability errors
   • Unknown use case errors
3. Clinical Rollout

Summary

• Powerful application framework
• Read access clinically available now
• Clinical write access coming
• Strong community and manufacturer support
• No Guidelines
Thanks