

Announcements 2215

Exam #2 THURSDAY 4/7 (login via Lab Machine)

DSO? <= let me know **before end** of class!

Cross Cutting Concerns

* Will look to each team to report/document

R2

Leave servers live and **readme** on how to access!

[Post mortems](#) + PeerEval ← Individual

(b4 Friday midnight)



Today

Lecture (XCC)

Activity



Cross Cutting Concerns in ERP

SWEN-343



Cross Cutting Concerns

Common functionality that spans multiple layers and tiers

Authentication

Authorization

Caching

Communication

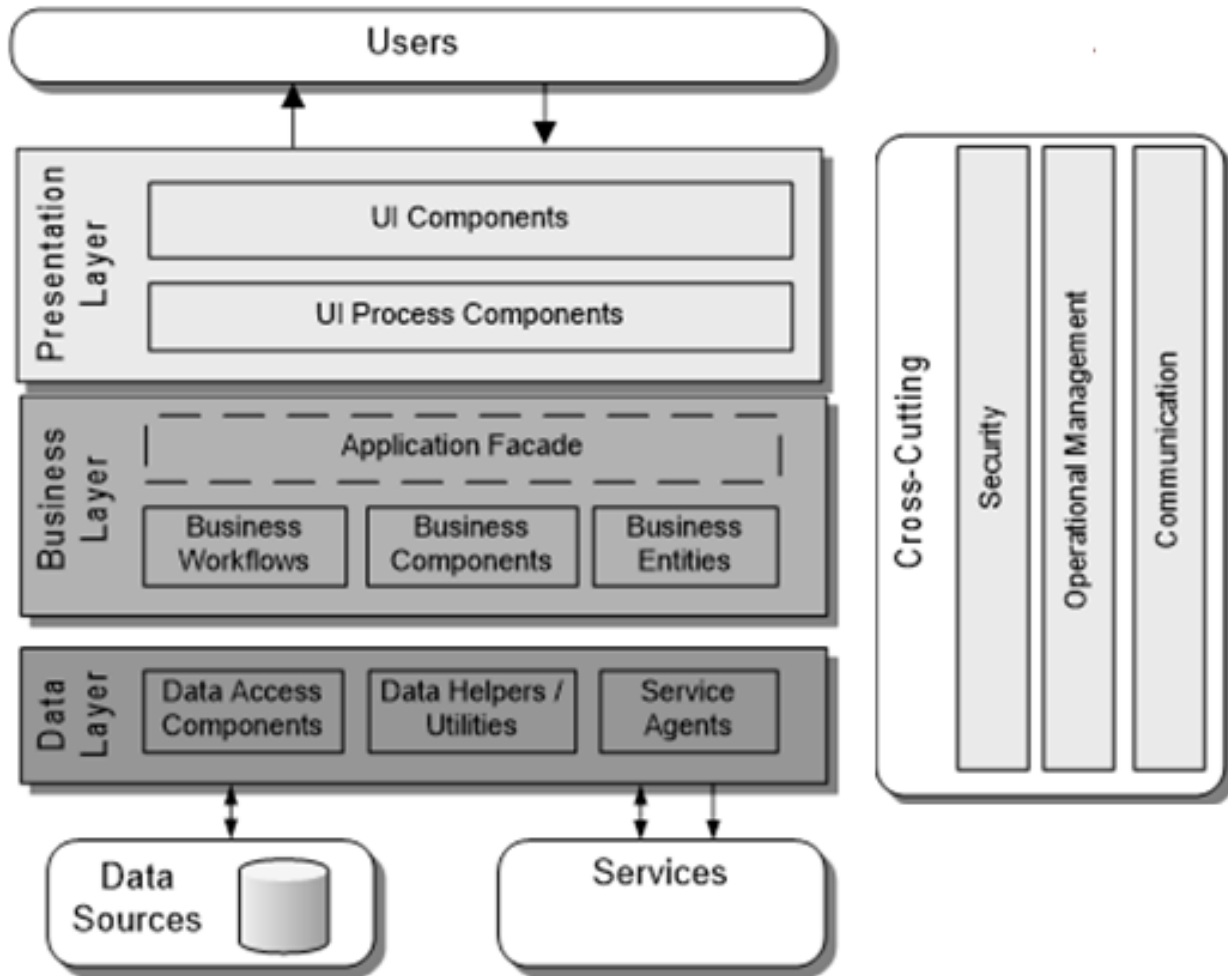
Exception Management

Logging

Validation

.... Many more....





XCC & ERP Systems - Concerns

How to properly design & implement system with XCC concerns?

Maintenance

Understandability

Extensionality

Functionality

Security



XCC & ERP Systems - Concerns

Who is in charge of managing XCC?

Who/how to put an appropriate **plan** in place?

How to **test** all XCC?

How to get an **agreement** on XCC?

How to deal with them?

How to test?

Etc.....



Other Concerns?



Specific Design Issues

Authentication

Authorization

Caching

Communication

Configuration Management

Exception Management

Logging

State Management

Validation



Authentication



Identify your **trust boundaries** and authenticate users and calls across the trust boundaries.

Enforce the use of strong passwords or password phrases.

If you have multiple systems within the application or users must be able to **access multiple applications** with the same credentials, consider a single sign-on strategy.

Do not transmit passwords **over the wire** in plain text, and do not **store passwords** in a database or data store as plain text. Instead, store a hash of the password.



Authorization



Role-based authorization for business decisions. Role-based authorization is used to subdivide users into **groups (roles)** and then set **permissions** on each role rather than on individual users.

Resource-based authorization for system auditing. **Resource-based** authorization sets permissions on the resource itself; for example, an access control list (ACL)

Identify your trust boundaries and authorize users and callers across the trust boundary.

Protect resources by applying authorization to callers based on their **identity, groups, or roles**. Minimize granularity by limiting the number of roles you use where possible



Caching

Do not cache volatile data, and do not cache sensitive data unless you encrypt it.

Do not depend on data still being in your cache; it may have been removed. Implement a mechanism to handle cache failures, perhaps by reloading the item from the source.

Be especially careful when accessing the cache from multiple threads. If you are using **multiple threads**, ensure that all access to the cache is thread-safe to maintain consistency.



Communication



Choose an appropriate transport protocol
HTTP, TCP etc..

Ensure that you **protect messages** and sensitive data during communication by using encryption, digital certificates, and channel security features.

If your messages do not need to be received in a specific order and do not have dependencies on each other, consider using **asynchronous** communication to avoid blocking processing or UI threads.



Exception Management



Design an appropriate **exception propagation strategy** that wraps or replaces exceptions, or adds extra information as required

Allow exceptions to bubble up to boundary layers where they can be logged and transformed as necessary before passing them to the next layer.

Ensure that the design deals with unhandled exceptions.

Ensure that a failure does not leave the application in an **unstable state**, and that exceptions do not allow the application to **reveal sensitive** information or process details.

Design an appropriate **logging and notification** strategy for critical errors and exceptions that stores sufficient details about the exception to allow support staff to recreate the scenario, but **does not reveal sensitive** information in exception messages and log files.



Logging

Design a **centralized** logging and instrumentation mechanism that captures **system- and business-critical** events.

Don't get too fine grained

Create secure log file management policies.

Do not store sensitive information in the log files,
and protect log files from unauthorized access.

Make logging useful



Main Factors for Managing XCC

Examine the functions required in each layer

Look for cases where you can **abstract** that functionality into **common components**.

It is likely that these kinds of components will be reusable in other applications.

Depending how you physically distribute the components and layers of your application, you **may need to install the crosscutting components on more than one physical tier**.

Still benefit from reusability and reduced development time and cost.



Dealing with XCC

Aspect Oriented Development

Common libraries

Consider using a **third-party library** of components that are highly configurable and can reduce development time.

May contain application blocks to assist with many concerns

Design Patterns

Decorator

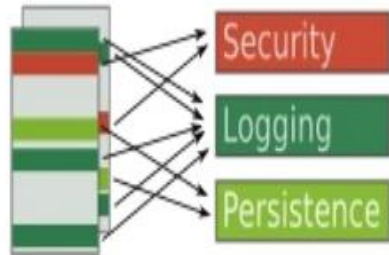
Dependency Injection



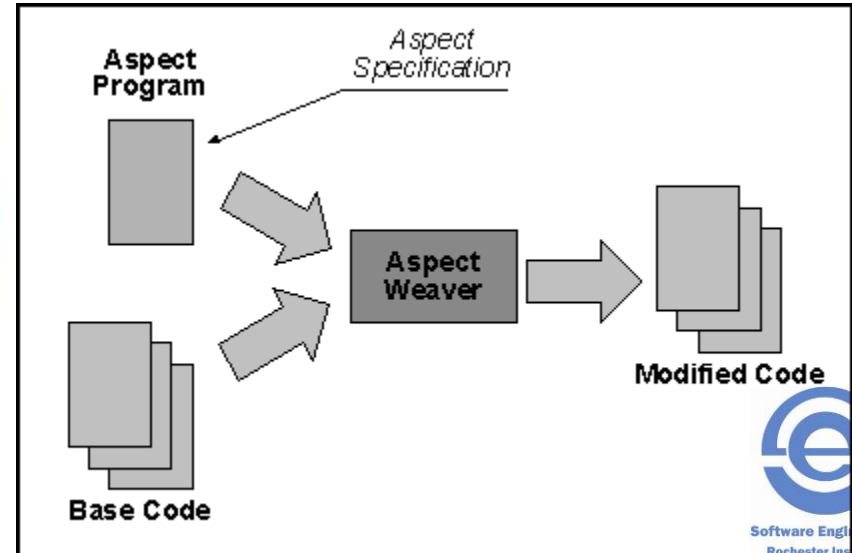
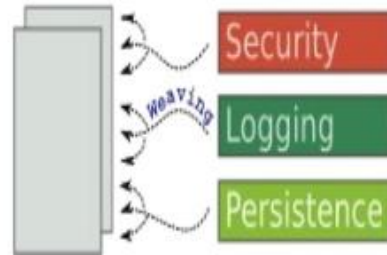
Aspect Oriented Development

Semi-popular way of dealing with XCC
“Weaves” code in during AOP compilation

Ye good ol way

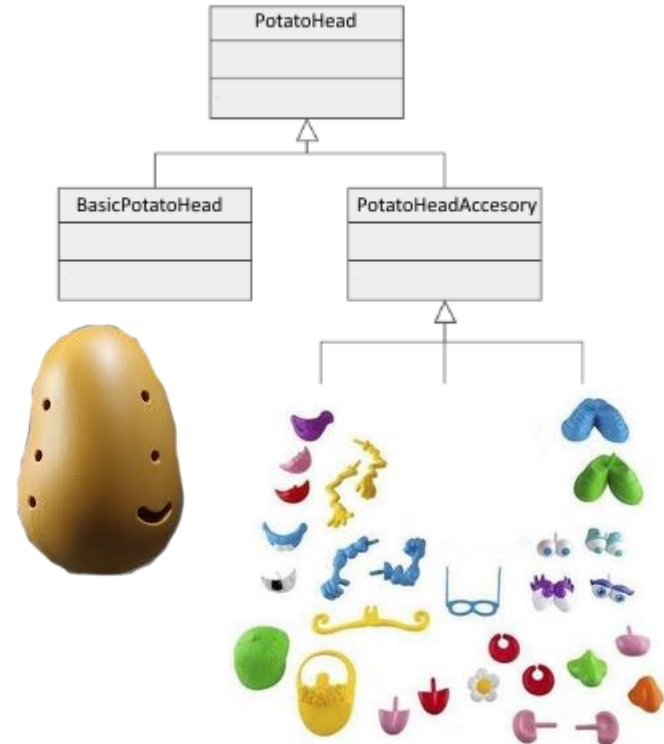


AOP way



Decorator Pattern

Add **new** functionality to an existing object without altering its structure.



Decorator Pattern

How can it help XCC?



Activity

Addressing and Planning for Cross Cutting Concerns

<http://www.se.rit.edu/~swen-343/activities/cross-cutting-concerns.html>



AUTHENTICATION	AUTHORIZATION	CACHING	COMMUNICATION	CONFIGURATION MANAGEMENT	EXCEPTION MANAGEMENT	LOGGING	STATE MANAGEMENT	VALIDATION

One List of your Silo and one list of Project as a whole

- Classify based on the above
- Green – already doing
- Red – stop doing
- Blue – start doing

The starfish technique uses five categories of issues.

Keep doing

These issues highlight an activity that worked well

No change necessary

More of

These issues request more of an activity

Start doing

These issues request the start of a new activity

Less of

These issues request less of an activity

Stop doing

These issues request stopping an activity that isn't serving the team, the product or the stakeholders

Starfish - 3 of 5

