

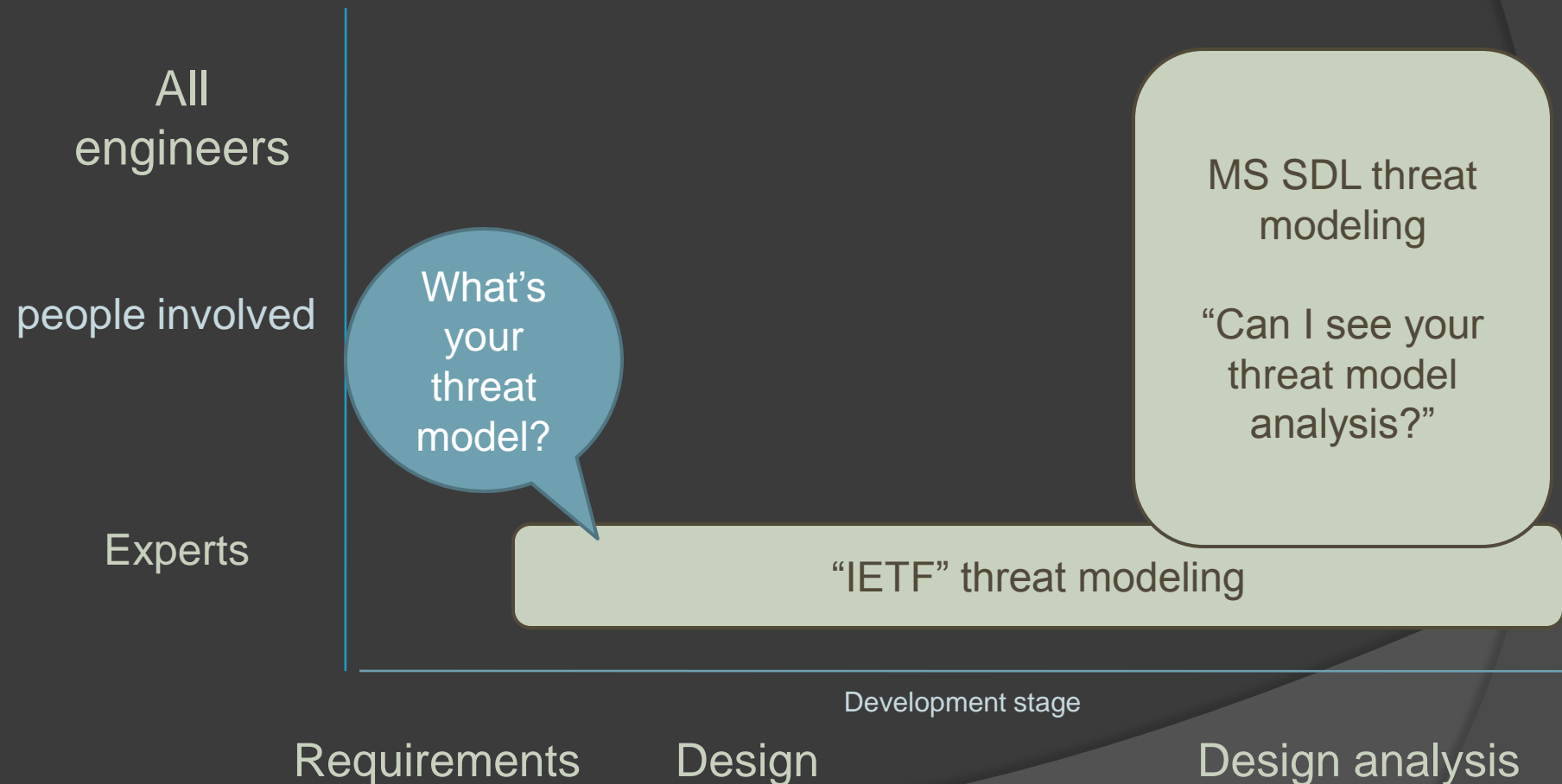
SDL THREAT MODELING: PAST, PRESENT AND FUTURE

Adam Shostack
Microsoft

Context: SDL

- Mandatory process at Microsoft
- Covers development from conception to shipping & updates
- Includes threat modeling during design phase

Terminology



THREAT MODELING:

PAST

MORDAC, THE PREVENTER
OF INFORMATION
SERVICES.

SECURITY IS MORE
IMPORTANT THAN
USABILITY.



www.dilbert.com
scottadams@aol.com

IN A PERFECT WORLD,
NO ONE WOULD BE
ABLE TO USE ANYTHING.



11-8-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.

To complete the
threat model
stare directly
at the sun.



© Scott Adams, Inc./Dist. by UFS, Inc.

Some history

- ⦿ Almost 10 years of SDL threat modeling
- ⦿ More than one process developed/year
- ⦿ Massive profusion of ideas and experiments

Process version history

- 1999 "Threats to Our Software" (Garms, Garg, Howard)
 - Developed STRIDE
- 2001 *Writing Secure Code* (Howard, LeBlanc)
- 2002 *Writing Secure Code*, 2nd edition (Howard, LeBlanc)
 - Wysopal/Howard work integrated @Stake, Microsoft processes
 - Added DREAD
- 2004 Formal rollout of security development lifecycle (SDL)
 - Includes threat model to meet secure-by-design commitment of SD3+C
- 2004 *Threat Modeling* (Swiderski, Snyder)
- 2006 *Security Development Lifecycle*, the book (Howard, Lipner)

Threat modeling issues

- ⦿ The process is complex
 - Eleven steps
 - " Only works with an expert in the room"
 - Jargon overload
- ⦿ The process is disconnected from development
- ⦿ "We're a component, we don't have assets"
- ⦿ Few customers for threat modeling artifacts
 - "Throw it over the wall to security"
- ⦿ It's hard to tell if the threat model is
 - Complete?
 - Accurate and up-to-date?
- ⦿ Expensive to do, value not always clear
 - (Especially if you're not sure how to threat model)
- ⦿ Training
- ⦿ The list of pain points goes on and on...

“The process that works for me is...”

- ◎ SDL process
- ◎ *Writing Secure Code* process (Howard and LeBlanc)
- ◎ *Threat Modeling* (Swiderski and Snyder, Microsoft Press)
- ◎ "Guerilla Threat Modeling" (Torr)
- ◎ Patterns and Practices (J.D. Meier)
- ◎ Threat modeling for dummies (Larry Osterman)
- ◎ Line-of-business threat modeling (ASAP/ACE team)
- ◎ Per team
 - MED threat modeling (Lyons)
 - "Creating High-Quality Shell TMAs" (Yadav, Sheldon, Douglas)

**SDL THREAT
MODELING:**

PRESENT

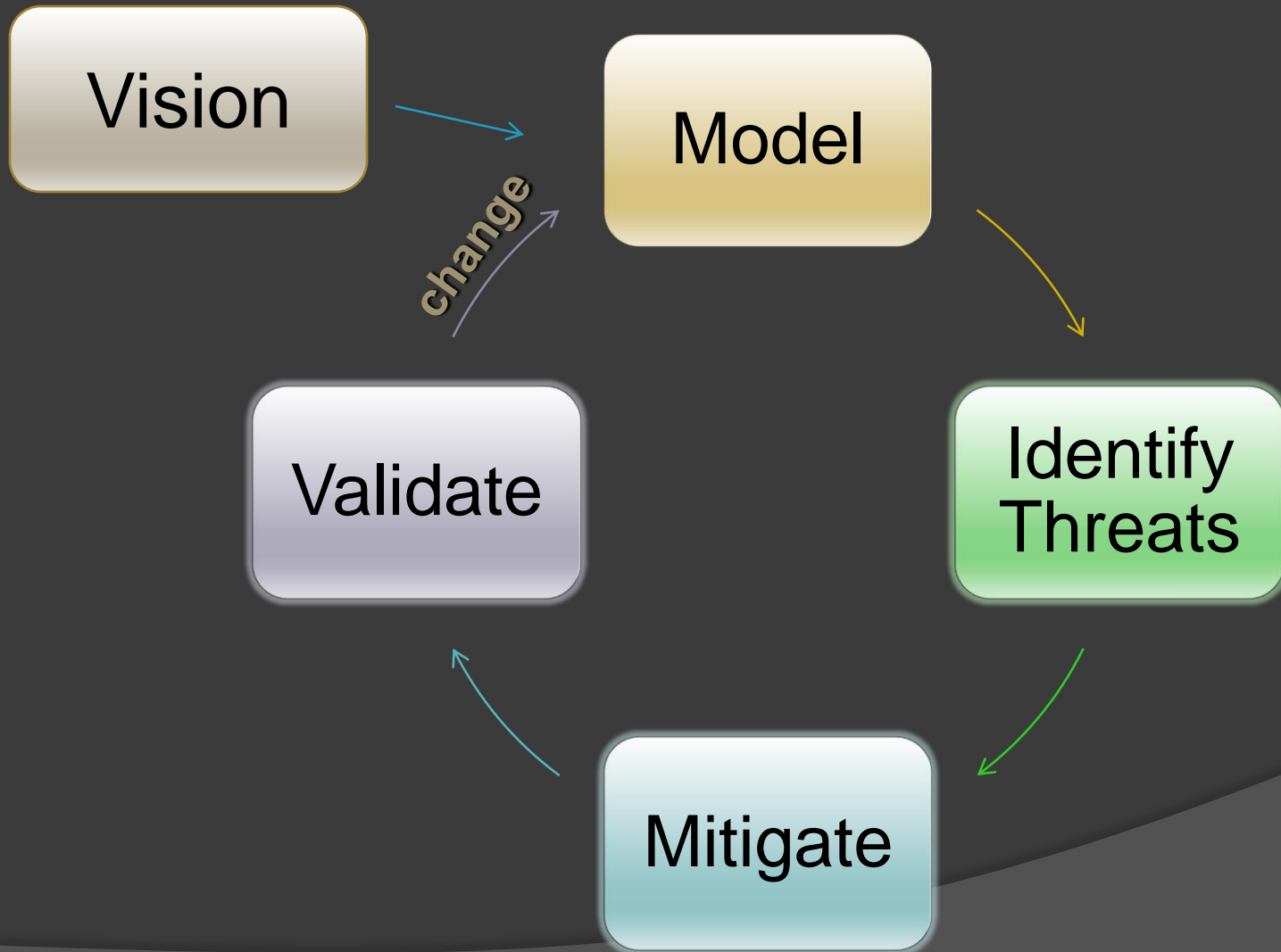
New SDL process addresses many issues

- The process is complex
 - Eleven steps
 - "Only works with an expert in the room"
 - Jargon overload
- The process is disconnected from development
- We're a component with no assets
- Few customers for threat modeling artifacts
 - "Throw it over the wall to SWI"
- It's hard to tell if the threat model is:
 - Complete?
 - Accurate and up-to-date?
- Expensive to do, value not always clear
 - (Especially if you're not sure how to threat model)
- Training
- Four-step process
- Explicit jargon purge
- Product studio integration
- TM based on software, not attacker
- TM as collaboration tool
- Self-checks in process
- Make it easier
- Threats as bugs
- Mitigations as features
- Better training

SDL TM process training

(Abridged)

Evolved SDL Process



Vision

Scenarios

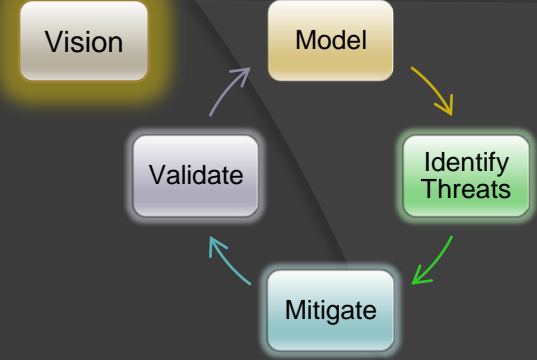
- Where do you expect the product to be used?
- Live.com is different from Vista
- MLB.com is different from an internal web site

Use cases/use Stories

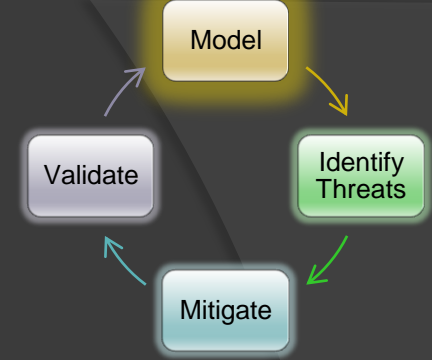
Add security to scenarios, use cases

Assurances

- Structured way to think about “what are you telling customers about the product’s security?”

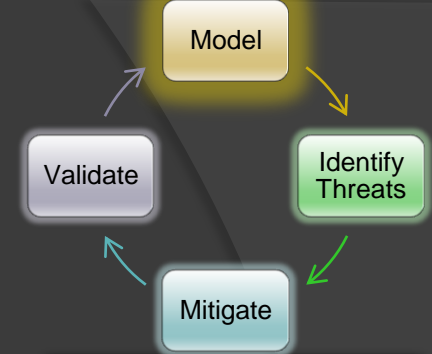


Model



- Create a software diagram
- Start with a overview which has:
 - A few external interactors
 - One or two processes
 - One or two data stores (maybe)
 - Data flows to connect them
- Check your work
 - Does it tell the story at an elevator pitch level?
 - Does it match reality?
- Break out more layers as needed

Diagram Elements



External entity

- People
- Other systems
- Microsoft.com
- etc...

Process

- DLLs
- EXEs
- COM object
- Components
- Services
- Web Services
- Assemblies
- etc...

Data Flow

- Function call
- Network traffic
- RPC
- Etc...

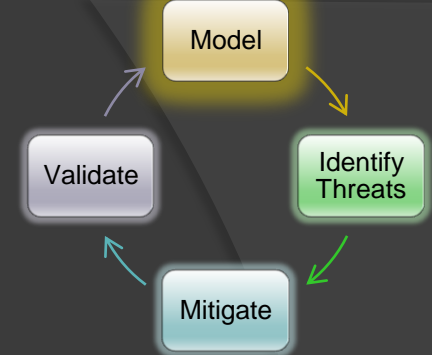
Data Store

- Database
- File
- Registry
- Shared Memory
- Queue/Stack
- etc...

Trust Boundary

- Integrity levels
- File system
- Session
- Network

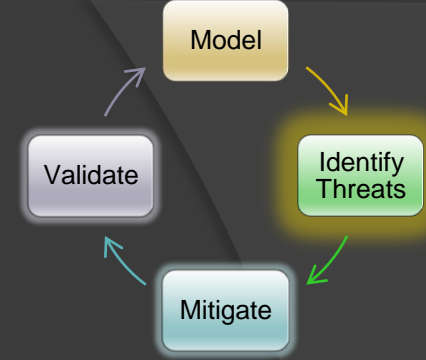
Creating Diagrams (2)



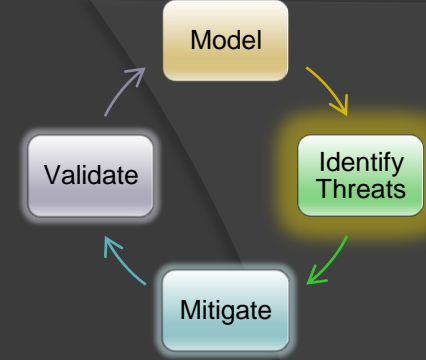
- Iterate over processes, data stores, and see where they need to be broken down
- How to know it “needs to be broken down?”
 - More detail is needed to explain security impact of the design
 - Object crosses a trust boundary
 - Words like “sometimes” and “also” indicate you have a combination of things that can be broken out
 - “Sometimes this datastore is used for X”...probably add a second datastore to the diagram

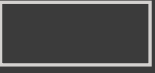



Identify Threats

- ⦿ Sounds good, but remember we're asking all engineers to be involved
- ⦿ How do you do it if you're not an expert?
- ⦿ Requires prescriptive guidance



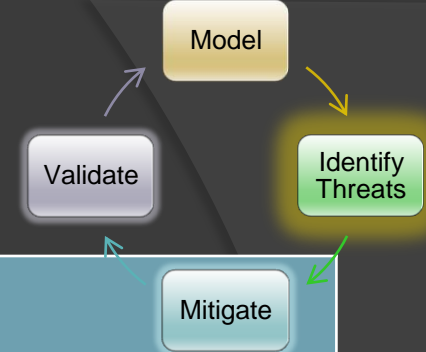
"STRIDE per Element"



	Spoofting	Tamper.	Rep.	Info.Disc.	DoS	EoP
 External Entity	✓		✓			
 Process	✓	✓	✓	✓	✓	✓
 Data Store		✓	✗	✓	✓	
 Dataflow		✓		✓	✓	

This is our chart; it may not be the issues you need to worry about

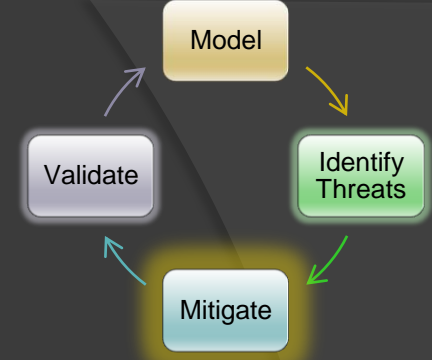
Threats & Properties



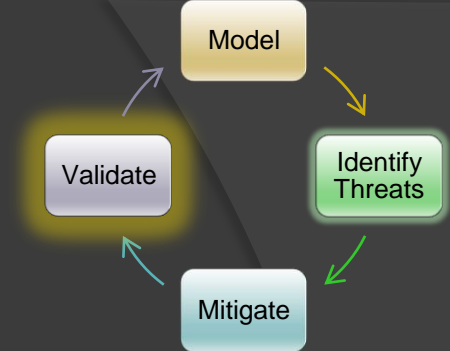
Threat	Property	Definition	Example
Spoofing	Authentication	Impersonating something or someone else.	Pretending to be any of billg, microsoft.com or ntdll.dll
Tampering	Integrity	Modifying data or code	Modifying a DLL on disk or DVD, or a packet as it traverses the LAN.
Repudiation	Non-repudiation	Claiming to have not performed an action.	"I didn't send that email," "I didn't modify that file," "I <i>certainly</i> didn't visit that web site, dear!"
Information Disclosure	Confidentiality	Exposing information to someone not authorized to see it	Allowing someone to read the Windows source code; publishing a list of customers to a web site.
Denial of Service	Availability	Deny or degrade service to users	Crashing Windows or a web site, sending a packet and absorbing seconds of CPU time, or routing packets into a black hole.
Elevation of Privilege	Authorization	Gain capabilities without proper authorization	Allowing a remote internet user to run commands is the classic example, but going from a limited user to admin is also EoP.

Mitigate Threats

- ⦿ Address each threat
- ⦿ Four ways to address threats:
 - Redesign to eliminate
 - Apply standard mitigations
 - Michael Howard's "Implementing Threat Mitigations"
 - What have similar software packages done?
 - How has that worked out for them?
 - Invent new mitigations
 - Riskier
 - Accept vulnerability in design
 - SDL rules about what you can accept
- ⦿ **Address each threat**



Validate



- ◎ Validate the whole TM
 - Does diagram match final code?
 - Are threats are enumerated?
 - Minimum: STRIDE per element that touches a trust boundary
 - Has test reviewed the model?
 - Tester approach often finds issues with TM, or details
- ◎ Is each threat mitigated?
 - Are mitigations done right
 - Examples are tremendously helpful here

Threat model at the right time

“Sir, we’ve analyzed their attack pattern and there is a danger”



SDL & BEYOND:

**THE FUTURE OF
THREAT MODELING**

Possibilities for the future

- ◎ Types of threat modeling
 - Asset-driven
 - Attacker-centric
 - Architecture-centric
 - Network protocol oriented
 - Others!
- ◎ Thinking about threat modeling
 - as a tool (mental toolbox)
 - Using tooling (software toolbox)

How do we assess these?

- ⦿ Let's think about tools/frameworks/orientations to help us think about security tools
- ⦿ The future is in better thinking about security tools
 - How do we assess and test the tools in our mental & software toolboxes?
- ⦿ Design as a framework for tradeoffs

Design Techniques

- ⦿ How and why to think about design
- ⦿ Usability for programmers
- ⦿ Flow

Design is about Tradeoffs

- ⦿ There is no ideal car
 - Market supports half dozen major manufacturers
 - Each has an extensive product line
 - That's mostly ignoring other modes of transport...bikes to busses to taxis to trains
 - There are car nuts and car haters
 - There's diversity of preferences, goals and budgets
- ⦿ Similarly, there is no one threat modeling process

Assessment: usability

strncpy and strncat — consistent, safe, string copy and concatenation.

*Todd C. Miller
University of Colorado, Boulder
Theo de Raadt
OpenBSD project*

strncpy() zero-fills the remainder of the destination string, incurring a performance penalty. Of all these issues, the confusion caused by the length parameters and the related issue of NUL-termination are most important. When we audited the OpenBSD source tree for potential security holes we found rampant misuse of strncpy() and strncat(). While not all of these resulted in exploitable security holes, they made it clear that the rules for using strncpy() and strncat() in safe string operations are widely misunderstood. The proposed replacement functions, strlcpy() and strlcat(), address these problems by presenting an API designed for safe string copies (see Figure 1 for function prototypes). Both functions guarantee NUL-termination, take as a length parameter the size of the string in bytes, and provide an easy way to detect truncation. Neither function zero-fills unused bytes in the destination.

Usability for programmers (1)

- It's not just your mom
- Programmers are people too

Usability for programmers (2)

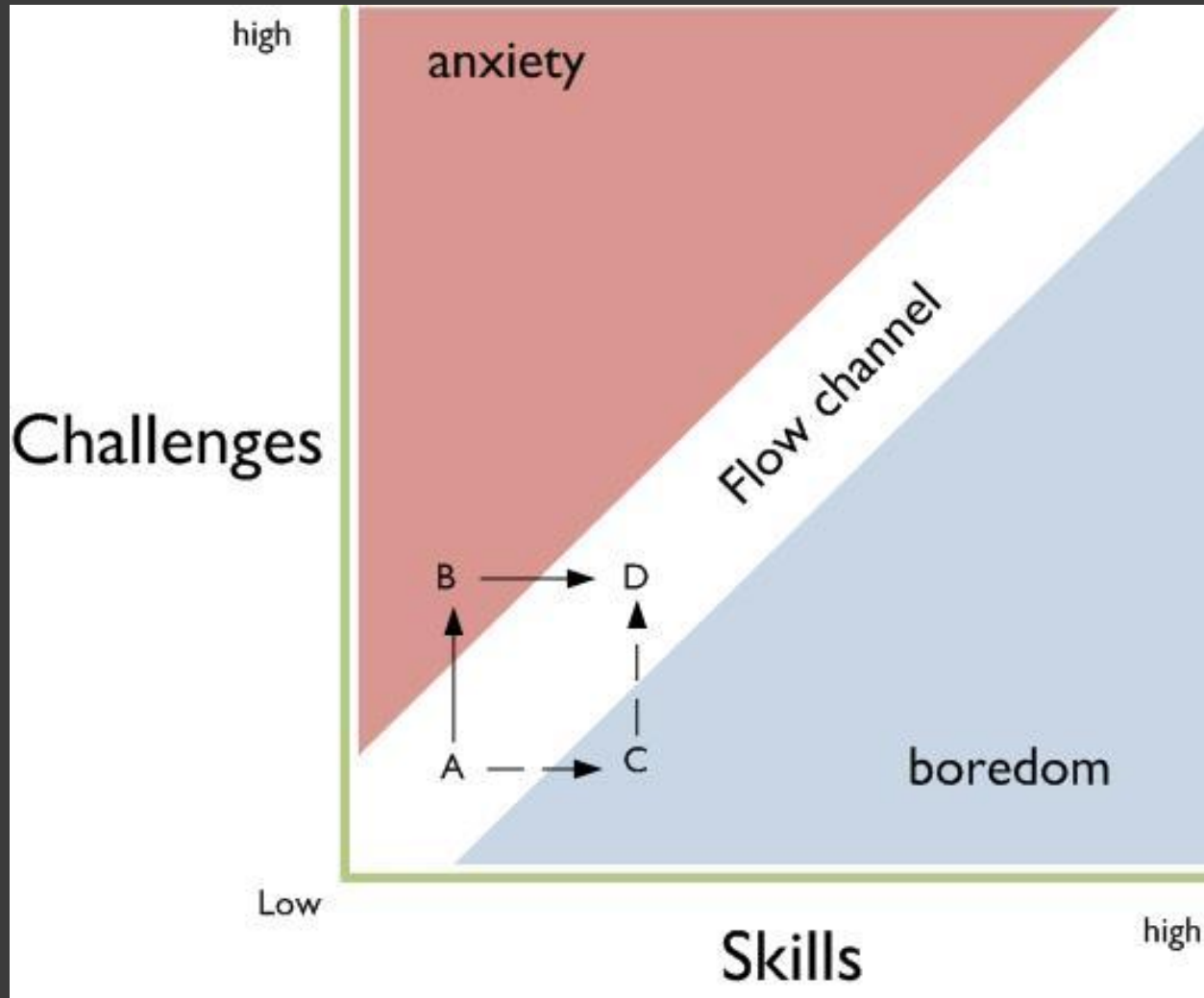
- ⦿ You don't need to be an expert to make usable software
 - It can help
- ⦿ Usability involves testing & iteration
 - “Paper prototypes”
 - “Think aloud”
 - Personas

Design Technique: Flow

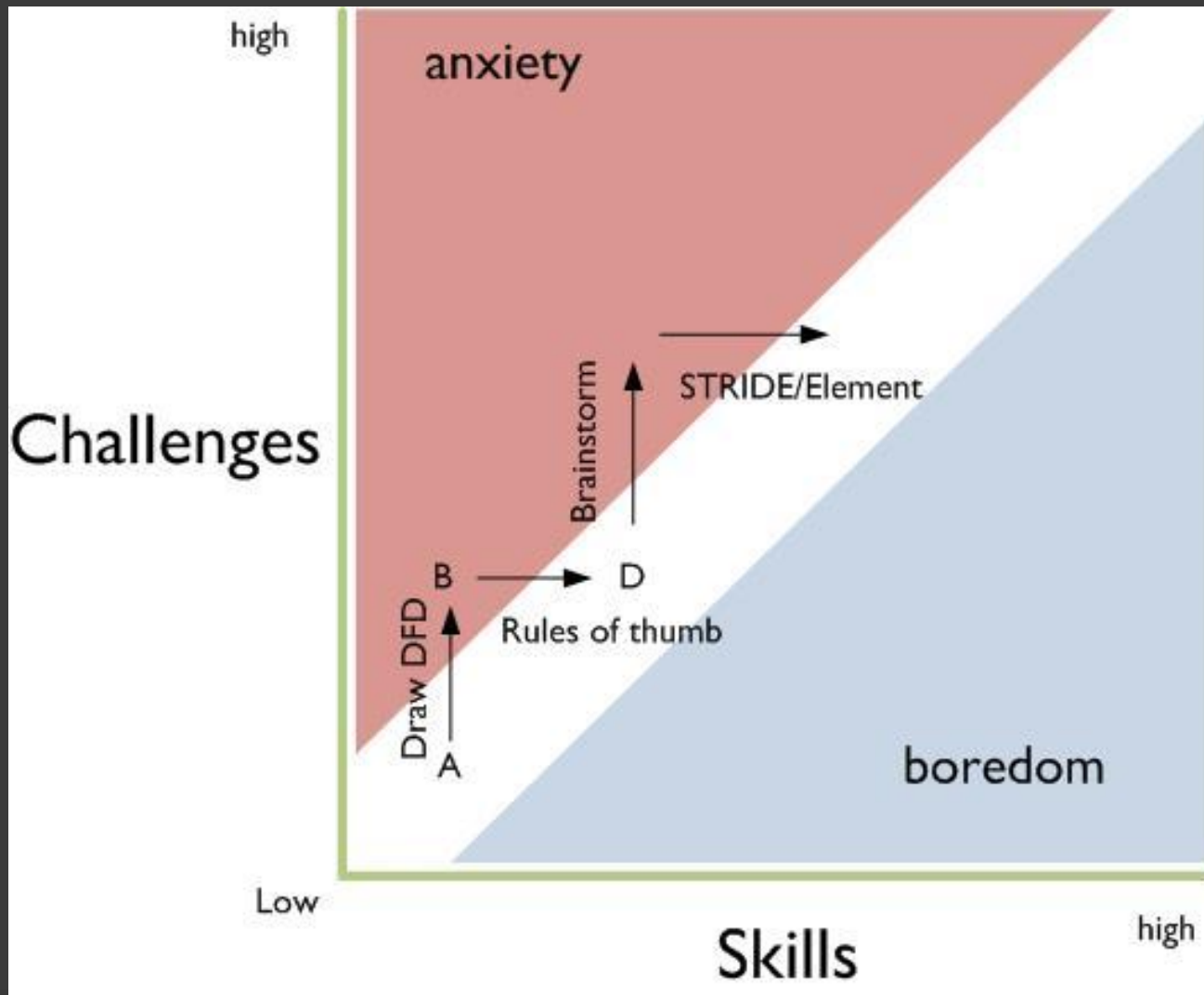
- ⦿ “the person is fully immersed in what he or she is doing, characterized by a feeling of energized focus, full involvement, and success”
- ⦿ Elements of flow (threat model issues highlighted)
 - Clear goals
 - Concentrating and focusing
 - A loss of the feeling of self-consciousness,
 - Distorted sense of time
 - Direct and immediate feedback
 - Balance between ability level and challenge
 - A sense of personal control over the situation or activity.
 - The activity is intrinsically rewarding
 - People become absorbed in their activity
- ⦿ **How the heck does this relate to threat modeling?**

Wikipedia: flow (psychology) or *Flow: The Psychology of Optimal Experience*.

Flow in a graph



Flow and threat modeling



What will the future hold?

Diverse Ecosystem of TM

- ⦿ Processes and tools which work for the problem at hand
- ⦿ Select one that will work for your project
 - Asset, attacker or software
 - Waterfall or agile
 - Experts or everyone
 - Firmware, boxed software, web, LoB, new devices, protocols, enterprises, etc
- ⦿ Modify it to work for your unique problems
- ⦿ Guidance from the philosophical to the prescriptive

Call to action

- ◎ Threat model!
- ◎ Enjoy yourself!
- ◎ Choose a system that works for your org
 - And threat model at the right time

- ◎ Read more: <http://blogs.msdn.com/sdl>

questions?

THANK YOU