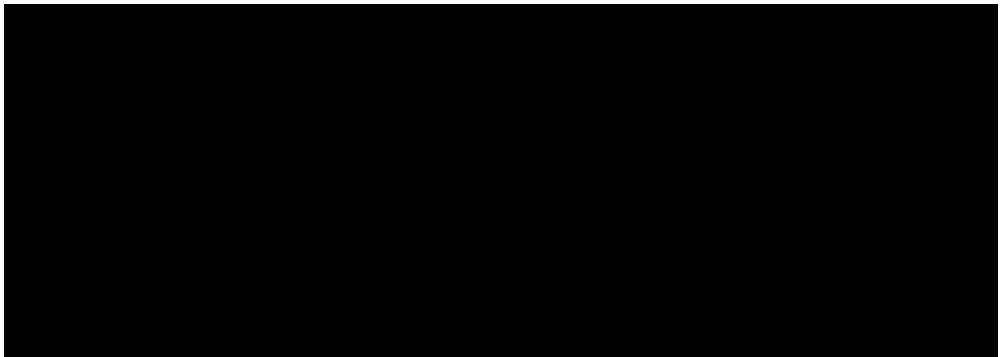
# Sécurité Applicative

OpenSAMM
Ve. 9 Jan. 2019 - PHELIZOT Yvan

# SAMM pour...

- **S**oftware
- **A**ssurance
- **M**aturity
- **M**odel

# Modèle de Maturité?



# Histoire

- 2008 : Open SAMM beta
- 2009 : Open SAMM 1.0
- 2017 : Open SAMM v1.5
- En cours: OpenSAMM v2

https://owaspsamm.org/

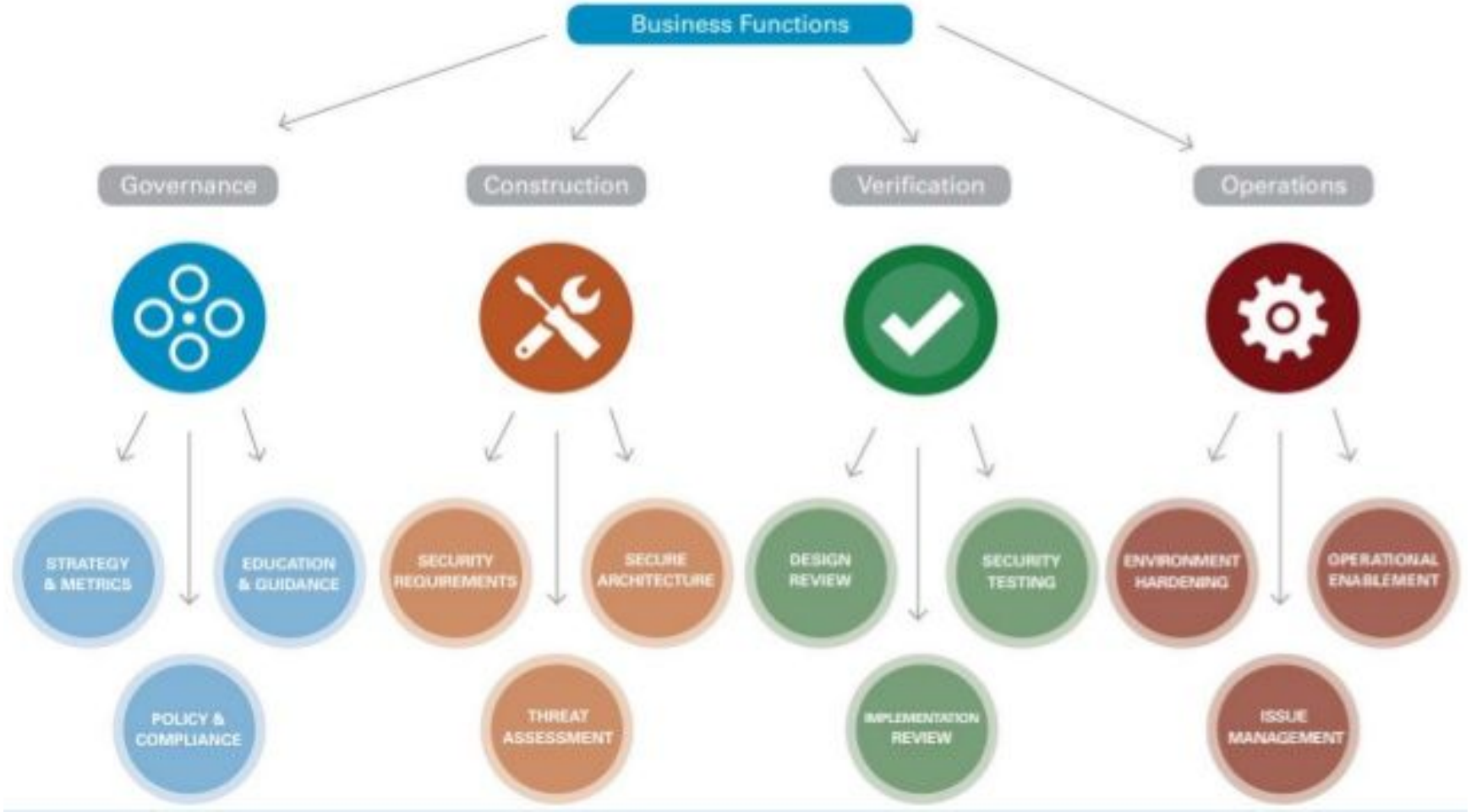
# But

- Evaluate an organization’s existing software security practices
- Build a balanced software security assurance program in well-defined iterations
- Demonstrate concrete improvements to a security assurance program
- Define and measure security-related activities throughout an organization

⇒ For all size of organisation

# Model

- Business function: Governance
- Security Practices: Strategy & Metrics
- Objective: Establish unified strategic roadmap for software security within the organization: Estimate overall business risk profile
- Activité: Estimate overall business risk profile
- Assessment: Is there a software security assurance program in place?
- Result: Concrete list of the most critical business-level risks caused by software


Business Functions
Governance
Construction
Verification
Operations
STRATEGY & METRICS
EDUCATION & GUIDANCE
POLICY & COMPLIANCE
SECURITY REQUIREMENTS
SECURE ARCHITECTURE
THREAT ASSESSMENT
DESIGN REVIEW
SECURITY TESTING
IMPLEMENTATION REVIEW
ENVIRONMENT HARDENING
OPERATIONAL ENABLEMENT
ISSUE MANAGEMENT

# Gouvernance

- Strategy & Metrics
  - Mettre en oeuvre le framework dans une organisation (roadmap)
  - Définir des buts mesurables et alignés avec les objectifs métier
  - Coût vs. Bénéfices
- Policy & Compliance
  - Identifier les exigences légales et réglementaires
  - Décliner ces exigences au niveau projet
  - Vérification des engagements au niveau projet
- Education & Guidance
  - Donner les ressources aux personnes pour produire un logiciel sûr
  - Former et certifier les personnes
  - Produire du contenu de référence

# Exemple d’activités

- Strategy & Metrics
  - Objectif SM2: Mesurer la valeur relative des actifs et choisir le niveau de tolérance au risques
    - Activité A : Classer les données et les applications selon leur risque métier
    - Activité B : Etablir et mesurer les buts en termes de sécurité selon cette classification
- Policy & Compliance
  - Objectif PC3 : Exiger la conformité et mesures les projets selons les règles de l’organisation
    - Créer des niveaux de conformité
    - Définir des solutions pour la collect des données d’audit
- Education & Guidance
  - Objectif EG1: Donner accès à des ressources autour du développement & déploiement sécurisé
    - Mettre en oeuvre des programmes de sensibilisation
    - Construire et maintenir des guides techniques

# Construction

- Threat Assessment
  - Identifier et comprendre les risques au niveau organisation & projet
  - Améliorer de manière incrémental le modèle
  - Mettre en oeuvre des réponses adaptées
- Security Requirements
  - Définir les exigences liées à la sécurité
  - Améliorer de manière incrémental les exigences
  - Assurer la prise en compte au niveau fournisseur
- Secure Architecture
  - Conseiller sur les architectures, logiciels et les pratiques sécurisés
  - Concevoir un logiciel sécurisé par conception

# Exemples d’activités

- Threat Assessment
  - Objectif TA1: identifier et comprendre les risques à un niveau macro
    - Construire et maintenir les modèles de menaces au niveau application
    - Construire le profil de l’adversaire/attaquant
- Security Requirements
  - Objectif SR2: Améliorer la granularité des exigences de sécurité à partir de la logique métier et des risques connus
    - Construire une matrice de contrôle d’accès (RBAC, …)
    - Lister les exigences à partir des risques connus
- Secure Architecture
  - Objectif SA1: considérer la sécurité en amont
    - Lister les frameworks logiciels recommandés
    - Appliquer les principes de sécurité à la conception

# Verification

- Design Review
  - Évaluer la sécurité de l'architecture et de la conception
  - Faciliter cette revue
  - Exiger la revue
- Implementation Review
  - Evaluer la sécurité au niveau logiciel et configuration
  - Identifier les problèmes connus (OWASP TOP 10, …)
  - Automatiser
- Security Testing
  - Evaluer la sécurité de manière dynamique
  - Définir des plans de tests pour les exigences de sécurité connues
  - Exiger les tests

# Exemples d’activités

- Design Review
  - Objectif DR2:proposer des services d’évaluation sur les bonnes pratiques de sécurité
    - Vérifier la complétude des mécanismes pour la sécurité
    - Proposer un service “à la demande” pour les projets
- Implementation Review
  - Object IR1: Lister les failles connues au niveau code
    - Créer des checklist pour les exigences connues
    - Vérifier les parties de code sensibles
- Security Testing
  - Objectif ST1: Mettre en oeuvre un processus pour tester la sécurité de l’application
    - Définir les cas de test à partir des exigences connues
    - “Pentester” l’application

# Operations

- Issue Management
  - Se préparer à gérer les anomalies et les incidents d’exploitation
  - Analyser les causes racines des problèmes
- Environment Hardening
  - Renforcer les environnements (Défense en profondeur)
  - Travailler en collaboration avec les équipes projet
  - Auditer la qualité de l’environnement
- Operational Enablement
  - Donner la capacité d’exploiter de manière sécurisée l’application
  -

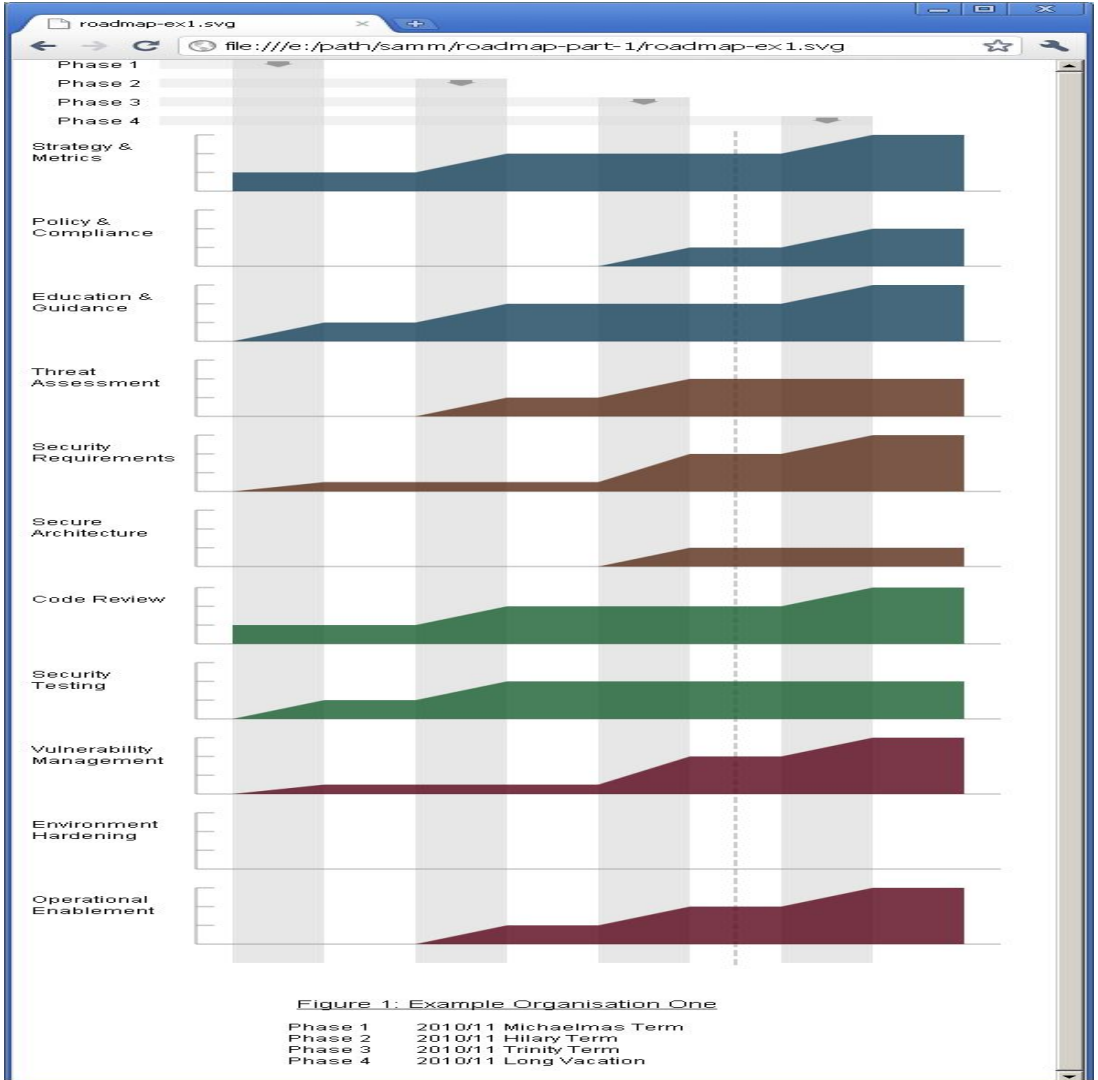
# Exemples d’activité

- Issue Management
  - Object IM1: Se préparer à gérer les anomalies et les incidents d’exploitation
    - Identifier les points de contact pour la remontée de failles
    - Créer une équipe pour répondre en cas d’alerte
- Environment Hardening
  - Objectit EH3 : Valider l’état de l’application
    - Identifier et déployer des outils de protection au niveau opérationnel
    - Etendre le programme d’audit à la configuration
- Operational Enablement
  - Object OE3: Rendre obligatoire la communication des informations de sécurité
    - Etendre le programme d’audit au dossier de transfert d’exploitation
    - Signer les composants d’application

# Maturity Levels

- Avant: un niveau ⇔ toutes les activités
- Après: prise en compte de chaque activité
- Rien n’empêche qu’une entreprise utilise des activités d’un niveau plus avancé

# RoadMap


roadmap-ex1.svg
file:///e:/path/samm/roadmap-part-1/roadmap-ex1.svg
Phase 1
Phase 2
Phase 3
Phase 4
Strategy & Metrics
Policy & Compliance
Education & Guidance
Threat Assessment
Security Requirements
Secure Architecture
Code Review
Security Testing
Vulnerability Management
Environment Hardening
Operational Enablement
Figure 1: Example Organisation One
Phase 1 2010/11 Michaelmas Term
Phase 2 2010/11 Hilary Term
Phase 3 2010/11 Trinity Term
Phase 4 2010/11 Long Vacation

# Evaluation

- Interview des personnes
- Scoring

# SAMM_Assessment_Toolbox_v1.5-Example_FINAL.xlsx

# BSIMM

| CODE REVIEW (CR) | | |
|---|---|---|
| ACTIVITY DESCRIPTION | ACTIVITY | PARTICIPANT % |
| **LEVEL 1** | | |
| Have SSG perform ad hoc review. | CR1.2 | 68.3 |
| Use automated tools along with manual review. | CR1.4 | 63.3 |
| Make code review mandatory for all projects. | CR1.5 | 33.3 |
| Use centralized reporting to close the knowledge loop and drive training. | CR1.6 | 36.7 |
| **LEVEL 2** | | |
| Assign tool mentors. | CR2.5 | 23.3 |
| Use automated tools with tailored rules. | CR2.6 | 16.7 |
| Use a top *N* bugs list (real data preferred). | CR2.7 | 20.8 |
| **LEVEL 3** | | |
| Build a factory. | CR3.2 | 3.3 |
| Build a capability for eradicating specific bugs from the entire codebase. | CR3.3 | 0.8 |
| Automate malicious code detection. | CR3.4 | 3.3 |
| Enforce coding standards. | CR3.5 | 2.5 |

# INSURANCE VS. HEALTHCARE VS. FINANCIAL SPIDER CHART

# BSIMM

- PT1.1: Use external penetration testers to find problems.
- SE1.2 : Ensure host and network security basics are in place.
- CMVM1.2: Identify software defects found in operations monitoring and feed them back to development.
- CP1.2: Identify PII obligations.
- AA1.1: Perform security feature review.
- SM1.4: Identify gate locations, gather necessary artifacts.


Governance
1. Strategy & Metrics (SM)
2. Compliance & Policy (CP)
3. Training (T)
Intelligence
4. Attack Models (AM)
5. Security Features & Design (SFD)
6. Standards & Requirements (SR)
SSDL Touchpoints
7. Architecture Analysis (AA)
8. Code Review (CR)
9. Security Testing (ST)
Deployment
10. Penetration Testing (PT)
11. Software Environment (SE)
12. Configuration Management & Vulnerability Management (CMVM)