THREATSVIEW
Make it Simple

# Enterprise Software House Replaces Email-and-Excel Pen-Test Chaos with a Structured Pre-Go-Live AppSec Workflow


## Client profile

A large financial services group with an in-house application development division building over 60 internal applications per year — core banking modules, branch tools, HR portals, customer onboarding apps, and partner integrations. Every application required a penetration test sign-off before production deployment.

## The challenge

The pre-go-live testing process was a nightmare every release cycle. Pen testers ran assessments and sent findings to developers as Word documents or Excel sheets attached to long email threads. Developers replied "fixed" inline in email, often without screenshots or evidence. Retesting was scheduled manually over Outlook calendar invites. Half the findings were missed during retest because there was no master list — testers worked off whatever was in their inbox. Release managers had no visibility into how many open findings existed per application, leading to last-minute production deployments with unresolved high-severity issues. Three different teams — AppSec, development, and release management — each maintained their own spreadsheet, and none of them matched. The release pipeline was being held hostage by Outlook threads.

## Why ThreatsView

The team needed a single platform where applications, developers, and pen testers all coexisted — where findings flowed directly from tester to developer without an email in between, where retesting was a structured workflow rather than a calendar invite, and where the release manager could see in one screen whether an application was clear to ship.

## Implementation highlights

***Asset onboarding***
All 60+ in-house applications onboarded into ThreatsView, each tagged with environment (dev/staging/UAT/prod), business criticality, data sensitivity, and target go-live date.

***People onboarding***
~140 developers across 18 product squads onboarded as users, each mapped to the applications they owned. Twelve internal pen testers onboarded as a separate role with finding-creation permissions.

***Pen tester workflow***
Testers log findings directly into the application's workspace — CVSS score, OWASP category, affected endpoint, proof-of-concept screenshots, reproduction steps. No Word. No email.

***Developer notifications***
The moment a finding is logged, the assigned developer receives an in-app notification plus email summary. The full finding — including evidence — is viewable in ThreatsView, not in an attachment.

***Retest workflow***
When a developer marks a finding as "Ready for Retest," the status flips on the tester's queue. The tester revalidates, attaches retest evidence, and either closes the finding or reopens it with comments. No manual scheduling required.

***Go-live gate***
The release management team gets a per-application dashboard showing open findings by severity. Production deployment is blocked at the CAB level for any application with open Critical or High findings.

# Enterprise Software House Replaces Email-and-Excel Pen-Test Chaos with a Structured Pre-Go-Live AppSec Workflow

## Results & impact

- Average finding lifecycle (logged → closed) dropped from 18 days to 5 days.
- Eliminated email threads for pen-test communication entirely — measured at zero finding-related emails after month two.
- 100% of pre-go-live findings now have documented evidence at both initial discovery and retest closure (previously closer to 40%).
- Release managers reported full visibility for the first time; “surprise” production deployments with open Highs dropped to zero.
- Pen-test team productivity increased — analysts ran ~35% more engagements per quarter without headcount change.
- Audit committee now receives a per-quarter application risk report generated directly from platform data.

> *“We used to negotiate go-live readiness over email. Now the platform just shows us whether the app is ready, and the conversation moves on.”*
>
> **— Head of Application Security**