

Staying current:

Building your upgrade
strategy





Ayesha Siddiqui Bradford

Senior Implementation
Consultant, Kindsight



**Caitlin
Kowaguchi**
Product Support
Manager, Kindsight



Emily Gee
Product Manager,
Kindsight



Cameron Aisenbrey

Director of Systems
Applications, Iowa State
University Foundation



Anida Campbell Ho
Sr. Director of Advancement
Business Solutions, Wisconsin
Foundation & Alumni Association



Welcome! Here's what we'll cover today:

Building the policy: What "good" looks like

Upgrades at Iowa State University Foundation

Upgrades at Wisconsin Foundation & Alumni Association

Breakouts

Final thoughts



The Reality: CRMs Evolve Multiple Times a Year



Cadence

- 3-4 major releases per year
- Regular patch updates
- Continuous security and feature delivery



Managed Packages

- You control upgrade timing
- Skipping releases builds compounding technical debt
- Releases build sequentially



Know Your Version

- Navigate to **Setup** in Salesforce
- Click **Installed Packages**
- Check your Ascend / AdvRM version number



The Stakes: The Cost of Falling Behind



The Cost of Falling Behind

- **Security & Compliance Risk:** Missing critical patches
- **Support Friction:** Legacy versions can add friction to and delay technical troubleshooting.
- **Features & Innovation Debt:** Paying for features you aren't using while building manual workarounds
- **Team Morale & Frustration:** Gift Processors might be stuck using outdated manual steps



The ROI of Proactive Upgrades

- **Predictable Windows:** Upgrades can be scheduled during low-activity periods
- **Minimize Disruptions:** Ensure that major year-end events or giving days are protected
- **Continuous Feature Adoption:** Immediate access to new out-of-box features without custom workarounds
- **Elevated Team Confidence:** Streamlined, modern UI and smooth user experience across all roles



Today's Outcome: Your Upgrade Blueprint



Partial Upgrade Policy Template

A customized blueprint defining:

- Upgrade cadence and timeframe
- Blackout windows
- Persona leads for functional testing



Prioritized Blocker & Action List

A clear roadmap identifying:

- Top technical and organizational hurdles
- Root causes (custom code, bandwidth, uncertainty)
- Specific action steps & Kindsight resources to solve them

Building Your Upgrade Policy: What does “good” look like?



What “Good” Might Look Like

Cadence & Windows

Fixed timeframe (ex. Within 45 days of a major release. Never within 30 days of FY close)

Clear Ownership

System Admin owns technical install.
Functional User owns workflow sign-off.

Standardized Testing

Defined scenarios for smoke and persona specific testing with pass/fail criteria in a designated Sandbox.

Change Communication

Proactive notices for CRM users before, during, and after upgrade.

Justifying Investment

Tying upgrade effort directly to solved end-user pain points and feature wins.

Rollback & Safety Net

Pre-planned rollback criteria, backups, and post-install validation logs.

Top 3 Upgrade Blockers (And How to Overcome Them)

“We have too many custom triggers, flows, and code.”

Kindsight Solution: Maintain a Pre-Release Inventory Log to monitor all custom items prior to upgrading.

“Our functional team doesn’t have time to test everything.”

Kindsight Solution: Prioritize testing 5-10 business-critical scenarios for key roles like Gift Processors, Stewardship, and Major Gift Officers rather than testing everything.

“We’re afraid of breaking gift processing during a major campaign.”

Kindsight Solution: Define policy-based blackout windows to prohibit upgrades.



IOWA STATE UNIVERSITY
FOUNDATION

Iowa State University Foundation: An Upgrade Strategy

Cameron Aisenbrey

Director of Systems Applications





About the Iowa State University Foundation

- Fundraising entity for Iowa State University
- Recently merged with Alumni Association
- Approximately 200 staff members
- Systems team consists of three admins and two developers
- Implemented first phase of Ascend in 2020
- Finished second phase in 2021





Staying Current

- **Initial concerns in early years**
 - Ran into several incidents where things broke after upgrade
 - Had to wait for support to fix and then upgrade to fix version
 - Led to lack of trust, staff hesitant to approve
- **Developed testing model**
 - Gift accounting team created spreadsheet of scenarios to test
- **Implemented SCRUM and structured timelines**
 - Two week sprints, three sprints to a release
 - Built in C sprint allows for dedicated upgrade time
- **Ascend releases improve reliability**
 - Have tightened up time to upgrade
 - No longer feel the need to let releases play out and address bugs
- **Developed plan to be as proactive as possible**
 - Pulled together structured upgrade plan for Ascend upgrades





Challenges/Nuances

- **Educating product owners**
 - How do we bring product owners, stakeholders up to speed on upcoming enhancements/fixes?
- **Gift accounting**
 - Testing plan
 - Requires sign off by several team members
- **Information management**
 - Takes time to test so many record types/scenarios
- **Time with environment refreshes/other product upgrades**
 - Refresh timeline – every 3 months
 - Have several other products that also upgrade – Blackthorn, Instant Address
- **Customizations**
 - We customize A LOT
 - Have to evaluate and double check JSON, metadata, settings, etc.
- **Potential Products**
 - What do we do with Kindsight products we don't use now but could use down the road?



Plan for Addressing Issues

- Meet with gift accounting every other week
- Testers submit user stories for issues they find when testing
- User stories are prioritized to keep upgrade moving
- Engage Kindsight Support if issues are not resolvable and ask for documentation regarding solution
 - Always start with Kindsight Support if we cannot resolve ourselves
 - Customizations in previous implementations affected - Start with support, move to implementer if necessary



Current State

- Have an Ascend Upgrade Strategy Plan
- Communication with stakeholders is key
- Stick to plan, but stay flexible with circumstances

Iowa State University Foundation Ascend for Salesforce Seasonal Upgrade Management Plan

Purpose

This document establishes a standardized process for evaluating, testing, approving, and implementing Kindsight Ascend releases within the Iowa State University Foundation Salesforce ecosystem. The objective is to implement new functionality as quickly as practical while minimizing risk to fundraising, gift processing, and year-end operations.



Wisconsin Foundation
& Alumni Association
UNIVERSITY OF WISCONSIN-MADISON

Wisconsin Foundation & Alumni Association

Anida Ho

Sr. Director of Advancement Business Solutions



About WFAA



**Non-profit that supports
University of Wisconsin-Madison**

700 users: (400) Staff & non-staff



Environment

Went live with Advancement RM Fall '23
2 full sandbox environments



4 Salesforce Admins

Plus a product lead & technical trainer
(developers are separate)



Governance group

Key system business SMEs



Cadence & Assessment

Today: one upgrade a year

Goal: two a year



Track known bugs

Log issues, note which upgrade resolves each one



Review upgrade notes

ID features/fixes we use, what needs testing, what can wait



Rate the impact

Score High / Medium / Low with business SMEs

Timeline Considerations



Map the timeline

Align with refreshes, Days of Giving, holidays & other upgrades



Test in sandbox first

Developer sandbox leveraging setup audit trail— see changes & estimate config time



Freeze decision

Decide if users can stay in the system or if it needs to freeze



Related systems impact

Integrations, data warehouse, custom solutions, etc

Testing



Template

Run reusable templates on critical functions, consider converted and new data



Upgrade-specific cases

Add upgrade and bug-fix tests



Related Systems & Integrations

Test end-to-end



Prepare, Test & Sign Off



Plan

Impact, timeline & resourcing



Communicate

Notify users, SMEs & leadership of dates & impact



Test

Scope the effort: critical & direct-impact functions



Go-live

Freeze decision, integration checks, sign-off

Breakouts

In your group, you'll discuss:

- Your current release against recent versions
- Gaps in your upgrade strategy
- Specific blockers (technical, org, resourcing) standing between you and staying current
- Strategies or processes you've put in place to overcome challenges



We're back!

- What are some of the standout blocker or insights you discussed?
- What's one concrete next step you'll be taking home?





Thank
you.