Comparison Report on Wishlist need vs Current Demo

---

1. Community Wishlist Survey 2022 vs Current Demo [Link]

Wishlist Requirement	In Demo?	Needed for Real Project	Notes/Limitations (if any)
Publish JS/CSS from public Git repo to wiki scripts/gadgets	Yes	Yes	
Auto-discover/select JS/CSS from repo	Yes	Yes	
Map each file to a wiki destination page	Yes	Yes	
Bulk (multi-file) publishing	Yes	Yes	
Authentication (BotPassword/OAuth)	Simulated	Actual API integration	Full authentication integration needed
Web GUI for all steps	Yes	Yes	
File preview before publishing	Yes	Yes	
Edit summary/metadata customization	Yes	Yes	
Sandbox/test mode	Yes	Yes	
Destination name auto-suggest/bulk prefix	Yes	Yes	
Linting/code standards (simulated)	Yes	Real linting	Actual code linting for production
Change diff viewer (pre-publish)	Yes	Yes	
Multi-wiki deployment	Yes	Yes	
Gadget meta-definition (dependencies/rights/settings)	Yes	Yes	

Localization/internationalization for gadgets	Yes	Yes	
Publishing log/history	Yes	Yes	
Smart sync: only publish changed files	Yes	Yes	
Community notification (simulated)	Yes	Real integration	Integration with actual discussion spaces
On-wiki review/approval process simulation	Yes	Real workflow	Real review/approval process integration as per future MW direction

---

2. Phabricator T36958 - User-Level Gadget Repositories (Gadgets 3.0)

User-Gadget Proposal/Feature	In Demo?	Needed for Real Project	Notes
Deploy scripts/gadgets to user namespace	Yes	Yes	Mapping to user pages/sandbox supported
Meta-definition (ResourceLoader JSON/Schema in userspace)	Partial	Full GadgetDefinition support	Demo supports meta-data for site gadgets; full .gadget JSON/schema in userspace is MediaWiki backend/platform task, not in scope for demo tool alone
Multi-file deployment	Yes	Yes	
User-to-user or cross-wiki gadget/module registration	No	No (unless core MW supports it)	Not possible unless MediaWiki Platform changes
Gadget enabling via Special:Preferences	No	No (handled by MW Platform)	For MediaWiki to implement

Dependencies, ResourceLoader rights/fields for gadgets	Yes	Yes	
Localization of gadget meta-data	Yes	Yes	
Creation/management of user-level .gadget pages	No	No (platform change required)	Not suitable for a deployment tool

---

3. Phabricator T71445 - Review Process for JS/CSS [Link]

Review/Workflow Requirement	In Demo?	Needed for Real Project	Notes
Deploy only reviewed JS/CSS to wiki	Yes	Yes	Assumes upstream (repo) review before publish
Attribution/audit log	Yes	Yes	
Collaborative review/fork/merge workflow on-wiki	Partial	Platform/extension needed	On-wiki fork/merge is a MW extension job, not handled by this demo/tool
Support for integration with CodeReview/FlaggedRevs	No	MW extension only	Needs MediaWiki extension, not applicable to deploy tool
GUI-based review/approve process	Simulated	Real workflow possible future	Demo simulates approval/review only
API limits, error handling, permissions	No	Yes	Real bot/tool must handle API/error/rights thoroughly

---

Summary Table: Present & Missing Features

Feature	Present (Demo)	Needed for Real Project	Not Present/Notes
Core publish workflow: repo → mapping → push	Yes	Yes	
Secure authentication (OAuth/BotPassword)	Simulated	Yes	Real implementation required
Sandbox/test deployment	Yes	Yes	
Edit summary/meta-data customization	Yes	Yes	
Gadget/property definition support	Yes	Yes	
Role/permission simulation	Simulated	Yes	Real rights check needed
Review/approval simulation	Yes	Yes	True on-wiki collaborative review is not implemented; would require platform changes/extensions
User-level gadget JSON/schema creation	No	No (for this tool)	Out of scope for deploy bot; MW backend only
Integration with Special:Preferences for gadgets	No	No	Platform only
Real API/error handling (timeouts, fails)	No	Yes	Needed in real production version
Notification/announcement real integration	No (simulated)	Yes (future)	Real integration with MW discussion spaces if desired

---

~User:Dev Jadiya

https://wikipublisher.vercel.app/