Calendar Solutions for SBNC Website

Technical Analysis & Comparison

Analysis Date: November 21, 2025
Workshop Date: November 18, 2025
Implementation Target: January 15, 2025
Research Basis: 22 user interviews, 67 pain points, 134 opportunities
Prepared For: Website Evaluation Team **Prepared by:**Ed Forman with extensive support from Claude, this loyal research assistant and ghostwriter.

Table of Contents

Executive Summary

The November 18 workshop identified the Events/Calendar page as a CRITICAL priority. The current Wild Apricot calendar gadget does not meet several minimum member requirements:

- Calendar position resets after viewing events (top complaint)
- No search functionality (members compare unfavorably to mobile app)
- Cannot filter events by type without admin creating multiple calendar pages
- Status indicators (Open/Full/Waitlist) require clicking into each event

This analysis compares eight calendar solutions across implementation approach (JavaScript vs. iframe), member requirements, development effort, brand customization, and maintainability.

Critical Use Case: Display Only

The calendar is for DISPLAY purposes only. The workflow:

1. Members browse events in the calendar interface
2. Members click an event to see details
3. Members go to Wild Apricot for registration and payment

This is NOT a registration system. The calendar does not need to:

- Handle payment processing

- Manage registration workflows
- Process waitlist logic
- Store member registration data
- Integrate Wild Apricot authentication

Implications:

- Calendar only reads event data from WA API (read-only, one-way data flow)
- No authentication, payment gateway, or database integration
- Calendar = event discovery and display; Wild Apricot = all transactions
- No sensitive payment or personal data handled by calendar
- Implementation estimates assume display-only scope

This display-only scope reduces technical complexity for JavaScript solutions and makes iframe solutions more viable since the two-step process (view calendar → register in WA) is built into the use case.

Solution Characteristics

Organizational Context	Solution Options	Key Attributes
Status quo maintained	Keep WA native (iframe)	Free, CSS customization possible, doesn't address all pain points
No JavaScript skills, minimal maintenance	TeamUp Calendar (iframe)	Hosted solution, no code maintenance, $8-20/mo
Technical volunteer available	FullCalendar.io (JS)	Most features, free, 4-6 days implementation
Budget for commercial support	Mobiscroll (JS)	Premium mobile experience, $500-1000/yr, vendor support

Solution Categories

Current Solution

1. Wild Apricot Native Calendar Gadget (current implementation)

JavaScript Solutions (Custom Integration)

Developer writes code connecting Wild Apricot API to calendar library for display. Reads event data from WA API, presents in calendar interface, links to WA for registration. Full control over features and design.

1. FullCalendar.io
2. DHTMLX Scheduler
3. Mobiscroll Event Calendar
4. Schedule-X

Iframe Solutions (Embedded Widgets)

Third-party service embeds via iframe. Minimal technical skills required. Limited customization.

1. TeamUp Calendar
2. Google Calendar Public Embed
3. Calendly / AddEvent

Side-by-Side Comparison: Core Capabilities

Requirements Fit (High Priority Features from Workshop)

Requirement	WA Native	FullCalendar	DHTMLX	Mobiscroll	Schedule-X	TeamUp	Google Cal
Clean, scannable layout	★★☆☆☆	★★★★★	★★★★☆	★★★★★	★★★★☆	★★★★★	★★★☆☆
Filter by event type	★☆☆☆☆	★★★★☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★★☆	★★☆☆☆
Show Open/Full/Waitlist	★☆☆☆☆	★★★★☆	★★★★☆	★★★★☆	★★★☆☆	★★★☆☆	★☆☆☆☆
Calendar + List views	★★☆☆☆	★★★★★	★★★★★	★★★★★	★★★★☆	★★★★★	★★★☆☆
Text search	☆☆☆☆☆	★★★★☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★★★	★★☆☆☆
Preserves position	☆☆☆☆☆	★★★★★	★★★★★	★★★★★	★★★★★	★★★★☆	★★☆☆☆
Mobile responsive	★★☆☆☆	★★★★★	★★★★☆	★★★★★	★★★★☆	★★★★☆	★★★☆☆
Custom brand colors/fonts	★★★☆☆	★★★★★	★★★★☆	★★★★★	★★★★☆	★★☆☆☆	★★☆☆☆
TOTAL	**10/40**	**35/40**	**31/40**	**34/40**	**30/40**	**31/40**	**18/40**

Cost & Effort Comparison

Solution	License Cost	Annual Cost	Dev Time	Maintenance	Risk Level
Current Solution					
WA Native	Included	$0	0 days	None	High (dissatisfaction)
JavaScript Solutions					
FullCalendar.io	Free (MIT)	$0	4-6 days	Medium	Low
DHTMLX Scheduler	$0-599	$0-599	7-11 days	Medium	Medium
Mobiscroll	$499-999	$499-999	4-6 days	Low (support)	Low
Schedule-X	Free (MIT)	$0	6-9 days	Medium-High	Medium
Iframe Solutions					
TeamUp Calendar	$0-20/mo	$0-240	0.5-1 day	None	Low
Google Calendar	Free	$0	1-2 days	Low	Medium
Calendly/AddEvent	$0-15/mo	$0-180	0.5-1 day	None	Medium

Development time estimates assume display-only scope (read WA API, show events, link to WA for registration). Times would be longer if calendar handled registration, payment, or authentication.

Brand Customization Capabilities

Solution	CSS Customization	SBNC Colors (Teal/Purple/Orange)	Fonts Control	Overall Rating
Current Solution				
WA Native	Yes (via Colors & Styles + CSS)	Yes (configurable)	Yes (via CSS)	★★★☆☆ Good
JavaScript Solutions				
FullCalendar.io	Complete control	Full control via CSS	Full control	★★★★★ Excellent
DHTMLX Scheduler	Complete control	Full control via CSS	Full control	★★★★☆ Very Good
Mobiscroll	Complete control	Full control via theming	Full control	★★★★★ Excellent
Schedule-X	Complete control	Full control via CSS	Full control	★★★★☆ Very Good
Iframe Solutions				
TeamUp Calendar	Limited (theme selection)	Limited theme colors	Limited	★★☆☆☆ Fair
Google Calendar	Minimal (basic colors only)	Very limited	No control	★★☆☆☆ Fair
Calendly/AddEvent	Minimal	Very limited	No control	★☆☆☆☆ Poor

Brand Customization Notes:

- **JavaScript solutions:** Full CSS control allows exact matching of SBNC brand colors (teal #4ECDC4, purple #7B2CBF, golden orange #FFB627) and fonts
- **WA Native:** Colors and Styles screen provides good customization; advanced CSS available via theme overrides for complete control
- **Iframe solutions:** Limited to vendor-provided themes; cannot precisely match brand guidelines

Maintainability & Support (Critical for Volunteer Organizations)

Factor	WA Native	FullCalendar	DHTMLX	Mobiscroll	Schedule-X	TeamUp	Google Cal
Core Library Maintenance							
Active			Yes (12+	Yes (10+	Yes (2+	Yes	Yes

development	Yes (WA)	Yes (15+ yrs)	yrs)	yrs)	yrs)	(vendor)	(Google)
Full-time team	Yes	Yes	Yes	Yes	Small	Yes	Yes
Release frequency	Per WA updates	Monthly	Quarterly	Monthly	Monthly	Continuous	Continuous
Security patches	WA-managed	Prompt	Prompt	Prompt	Prompt	Vendor-managed	Google-managed
Community Support							
Stack Overflow questions	500+	7,500+	2,000+	800+	200+	N/A (vendor)	50,000+
Documentation quality	Fair	Excellent	Good	Good	Fair	Excellent	Good
Tutorial availability	Limited	Extensive	Moderate	Moderate	Limited	Good	Extensive
GitHub issues response	N/A	Active	Active	N/A (closed)	Active	N/A (closed)	N/A
Commercial Support							
Paid support available	Yes (WA support)	Yes (~$200-500)	Yes ($599+)	Yes (included)	No	Yes (included)	No
Implementation consulting	Limited	Yes	Yes	Yes	No	Yes	No
Priority bug fixes	Standard queue	Yes (paid)	Yes (paid)	Yes	No	Yes	No
SLA guarantees	No	No	No	Yes	No	Yes	No
Non-Developer Maintainability							
Config-driven setup	Yes	Yes	Partial	Partial	No	Yes	Yes
Visual admin interface	Yes	No	No	No	No	Yes	Yes
Requires JS debugging	No	Yes	Yes	Yes	Yes	No	No
Error message clarity	Fair	Good	Fair	Good	Fair	N/A (hosted)	N/A (hosted)
Volunteer Handoff							

Risk							
Learning curve (new vol)	30 min	1-2 days	2-3 days	1-2 days	2-4 days	1 hour	30 min
Code complexity	None	Moderate	High	Moderate	Moderate	None	Minimal
Custom code volume	0 lines	500-800 lines	800-1200 lines	500-800 lines	600-1000 lines	0 lines	50-100 lines
Can non-tech admin update	Yes	No	No	No	No	Yes	Partial
Long-Term Viability							
Established track record	Ongoing	15 years	12 years	10 years	2 years	8 years	20+ years
Commercial backing	Yes	Yes	Yes	Yes	No	Yes	Yes
User base size	Large (WA users)	Very large	Large	Medium	Growing	Medium	Massive
Risk of abandonment	Low	Very low	Low	Low	Medium	Low	Very low
MAINTAINABILITY SCORE	**Good**	**Good**	**Fair**	**Good**	**Fair**	**Excellent**	**Good**

JavaScript solutions require ongoing technical volunteer availability. Iframe solutions trade customization for zero maintenance burden. WA Native offers zero maintenance but limited features for addressing workshop-identified pain points.

Detailed Solution Analysis

1. Wild Apricot Native Calendar Gadget (Current Solution)

Implementation Approach: Iframe widget embedding Wild Apricot system pages

Product Information: - Documentation: https://gethelp.wildapricot.com/en/articles/412-event-calendar-gadget - Theme Customization: https://gethelp.wildapricot.com/en/articles/438-css-customization - Support: https://support.wildapricot.com

Current State Assessment

This is SBNC's existing solution. Workshop identified it as failing to meet minimum requirements.

Member Requirements Coverage

Workshop Requirement	WA Native Performance	Rating
Clean, scannable calendar	Inefficient layout, wasted space	★★☆☆☆
Filter by event type	Admin creates separate calendar pages; not user-facing	★☆☆☆☆
Show Open/Full/Waitlist	Requires clicking into each event	★☆☆☆☆
Calendar + List views	Both exist but navigation clunky	★★☆☆☆
Text search	Not available	☆☆☆☆☆
Preserve position	Returns to top/beginning - architectural limitation	☆☆☆☆☆
Mobile responsive	Works but significant usability issues	★★☆☆☆
Custom brand colors/fonts	Customizable via Colors & Styles + CSS	★★★☆☆

Wild Apricot Platform Capabilities Research

Available Customization Features:

Colors and Styles Screen:

- Event calendar title typography (theme-dependent)
- Background colors and info boxes
- Event calendar-specific styling elements
- Changes apply across gadgets using same settings

CSS Customization:

- Full CSS customization available for advanced styling
- Custom CSS code can be entered in Advanced section of gadget settings
- Theme-specific CSS modifications supported

Theme Overrides:

- Complete HTML and CSS control for experienced web developers
- Can create custom themes or modify existing ones
- More powerful than previous CSS-only customization

Event Color-Coding:

- Events can be color-coded by event tags using custom JavaScript
- Requires adding custom script to website
- Allows visual differentiation of event types in calendar view

Event Filtering:

- Can filter events by event tags
- Multiple event calendars can display different event subsets
- Filtering options set in gadget configuration

View Options:

- List view
- Month view
- Week view (availability varies by theme)
- Day view (availability varies by theme)

Platform Limitations vs. Configuration Issues

Confirmed Platform Limitations:

- **Position Loss:** Architectural limitation of iframe-based gadget - cannot maintain scroll position after navigating to event detail and back. This is inherent to the iframe implementation and cannot be resolved through configuration.
- **No Built-in Search:** Event calendar gadget does not include native search functionality
- **User-facing Filtering:** While admin can create multiple filtered calendars, individual users cannot dynamically filter a single calendar view by event type

Potential Configuration Improvements:

- **Brand Colors:** Current dated appearance may be due to theme selection or lack of CSS customization rather than platform constraints
- **Event Status Display:** Custom fields and JavaScript could potentially show Open/Full/Waitlist without clicking (requires developer)
- **Event Color-Coding:** Event types can be visually distinguished using color-coding via custom JavaScript
- **Mobile Responsiveness:** May be improved through theme selection or CSS modifications

Member Pain Point Mapping

Every top-5 pain point relates to current WA calendar:

- "Difficulty navigating website compared to app" - Root cause is WA calendar limitations
- "Lack of search function" - WA gadget has no search
- "Inability to find previously signed-up events" - Separate dashboard issue but calendar contributes
- "Website design outdated and unappealing" - WA iframe styling limited, though CSS customization available
- "Calendar navigation loses position" - Core WA gadget behavior, cannot fix (platform limitation)

Maintenance & Support Profile

Zero Maintenance:

- Included with Wild Apricot subscription
- No code to maintain
- No developer needed for basic usage

The "Bus Factor":

- Zero risk for basic operation - any admin can manage
- CSS customization requires web development knowledge

Support:

- Wild Apricot support team available
- Cannot fix core navigation issues (by design/platform architecture)

Why Workshop Identified This as Critical Problem

From research presentation:

Lens 1: Pain Points (67 identified)

- Top 5 critical issues all relate to calendar/events

Lens 7: Poignant User Quotes

- “You click on something, and then you go back, and you’re back at the top of the calendar”
- “Each time I want to go back to the calendar of events… it goes back to the beginning”
- “I do find the calendaring part clunky. It seems antiquated”
- “The app has a search bar, but on the website, it doesn’t”

Lens 6: User Journeys

Every journey stage has friction:

1. Discovery → Overwhelmed
2. Exploration → No search, manual scrolling
3. Evaluation → Loses calendar position, unclear capacity
4. Registration → Hard to find “Join” button
5. Post-Registration → Cannot find “My Events”

Cost-Benefit Analysis

Benefits:

- $0 cost
- $0 maintenance effort (for basic usage)
- Already implemented
- Familiar to current administrators
- CSS customization available for brand matching

Trade-offs:

- Cannot address critical member pain points within WA platform limitations
- Position loss after event click (architectural limitation of the gadget)
- No search functionality
- Limited interactive filtering capabilities
- Iframe-based implementation affects mobile responsiveness

- Layout inefficiencies documented in user feedback
- Members express preference for mobile app over website calendar
- CSS customization requires developer knowledge

Brand Customization:

- **Rating:** ★★★☆☆ (Good)
- Colors and Styles screen provides straightforward customization
- Advanced CSS customization available for exact brand matching
- Theme overrides offer complete control for experienced developers
- Current “dated” appearance may be configuration issue rather than platform constraint
- SBNC brand colors (teal, purple, golden orange) can be applied via CSS

JavaScript Solutions: Detailed Analysis

2. FullCalendar.io

Implementation Approach: JavaScript library integrated with Wild Apricot API (display-only)

Product Information: - Website: https://fullcalendar.io - Documentation: https://fullcalendar.io/docs - GitHub: https://github.com/fullcalendar/fullcalendar - License: MIT (open source) - Commercial Support: https://fullcalendar.io/support

Member Requirements Coverage

Workshop Requirement	Implementation	Member Impact
Clean, scannable calendar	Google Calendar-style interface	Familiar, reduces learning curve
Filter by event type	Custom dropdown with API queries	“Show me only Tennis events” works
Show Open/Full/Waitlist	Color-coded events, status badges	No clicking into full events
Calendar + List views	Native Month/Week/Day/List modes	Mobile users prefer List view
Text search	Custom search box filters events	Matches mobile app capability
Preserve position	Single-page app maintains scroll	Fixes #1 complaint
Mobile responsive	Touch-friendly, adapts to screen	Works well on phones
Custom brand colors/fonts	Complete CSS control	Exact SBNC brand matching

Brand Customization

Rating: ★★★★★ (Excellent)

Capabilities:

- **Complete CSS control:** Every visual element can be styled
- **SBNC Brand Colors:** Can precisely match teal (#4ECDC4), purple (#7B2CBF), golden orange (#FFB627)
- **Typography:** Full control over font families, sizes, weights
- **Event styling:** Custom colors for event types, status indicators
- **Responsive design:** Brand consistency across all devices

Implementation:

```
/* SBNC Brand Colors */
:root {
  --sbnc-teal: #4ECDC4;
  --sbnc-purple: #7B2CBF;
  --sbnc-orange: #FFB627;
}

.fc-event-tennis { background-color: var(--sbnc-teal); }
.fc-event-wine { background-color: var(--sbnc-purple); }
.fc-event-social { background-color: var(--sbnc-orange); }
```

Maintenance & Support Profile

Core Library Health:

- Repository: github.com/fullcalendar/fullcalendar
- Stars: 18,000+ | Forks: 3,500+ | Contributors: 200+
- Last release: Within 30 days (active development)
- License: MIT (permissive open source)
- Creator: Adam Shaw (actively involved since 2009)
- Commercial entity: FullCalendar LLC provides paid support

Finding Help When Problems Occur:

- Stack Overflow has 7,500+ answered questions
- Official documentation includes interactive examples
- Community forum at fullcalendar.io/community
- GitHub issues typically responded to within 2-3 days

The “Bus Factor” Reality:

If your technical volunteer becomes unavailable, next volunteer faces:

- Code to understand: ~500-800 lines of SBNC-specific JavaScript
- Documentation needed: Clear comments on Wild Apricot API integration
- Skills required: Basic JavaScript, understanding of APIs
- Time to onboard: 1-2 days reading code + documentation
- Finding replacement: Many developers know FullCalendar

Mitigation strategies:

- Document all SBNC customizations in separate README

- Keep WA API integration code isolated in dedicated files
- Use configuration objects (JSON) for event type mappings
- Consider one-time $300-500 professional code review/documentation

When you need emergency help:

- FullCalendar LLC offers incident-based support ($200-500)
- Freelance developers on Upwork (search “FullCalendar”) ~$50-100/hr
- Many local developers have FullCalendar experience

Development Estimate

These estimates are for display-only implementation (reading WA API and showing events). Does not include registration/payment handling, which remains in Wild Apricot.

Phase 1: Core Calendar (2 days)

- Initialize FullCalendar with basic configuration
- Connect to Wild Apricot API for event data (read-only)
- Display events in calendar view
- Link events to Wild Apricot registration pages
- Basic navigation (month/week/day)

Phase 2: Filtering & Search (2-3 days)

- Build event type filter dropdown
- Implement text search functionality
- Add date range picker
- Display status indicators (Open/Full/Waitlist based on WA data)

Phase 3: Polish & Mobile (1-2 days)

- Apply SBNC brand colors and fonts
- Mobile responsiveness testing
- Performance optimization
- Browser compatibility testing
- User acceptance testing

Total: 5-7 days developer time (display-only scope)

Member Pain Point Resolution

Member Quote	How FullCalendar Solves It
“You click something and go back, you’re back at the top”	Single-page app maintains scroll position
“The app has a search bar, but website doesn’t”	Custom search implementation possible
“Helpful to know which events still have openings”	Status badges show before clicking
“I do find the calendaring part clunky, antiquated”	Modern UI with smooth interactions
“I just want to do my business and get out”	Faster task completion with filters

Pros for SBNC

- Free and open-source (no ongoing license fees)
- Most mature and proven solution (15 years)
- Largest community (easiest to find help)
- Extensive documentation reduces learning curve
- Can implement incrementally (basic first, enhance later)
- Future volunteers likely to have heard of it
- Complete brand customization control

Cons for SBNC

- Requires JavaScript developer for initial build
- Custom code needs documentation for future volunteers
- If developer leaves, need replacement with JS skills
- No visual admin interface (code changes required)
- Support available but costs extra

3. DHTMLX Scheduler

Implementation Approach: Enterprise-grade JavaScript scheduling library (display-only)

Product Information: - Website: https://dhtmlx.com/docs/products/dhtmlxScheduler/ - Documentation: https://docs.dhtmlx.com/scheduler/ - GitHub: https://github.com/DHTMLX/scheduler - License: GPL v2 (free) or Commercial ($599+) - Pricing: https://dhtmlx.com/docs/products/dhtmlxScheduler/#licensing

Member Requirements Coverage

Similar to FullCalendar but with more complex configuration. Strong accessibility features (WAI-ARIA, keyboard navigation). May overwhelm users who prefer simplicity over feature richness.

Brand Customization

Rating: ★★★★☆ (Very Good)

Capabilities:

- Complete CSS control for styling
- SBNC brand colors can be precisely applied
- Theme system with customizable elements
- Font control available
- Slightly more complex configuration than FullCalendar

Implementation Consideration:

- More configuration options may require additional styling decisions
- Enterprise focus means solid but complex theming system

Maintenance & Support Profile

Core Library Health:

- Repository: github.com/DHTMLX/scheduler (commercial, limited public repo)
- Documentation: Thorough but dense
- License: GPL v2 (free) or Commercial ($599+)
- Company: DHTMLX (15+ years in business)

Support Characteristics:

- Community forum exists but smaller than FullCalendar
- Stack Overflow: ~2,000 questions (adequate but less than FullCalendar)
- Commercial support included with paid license
- Documentation thorough but assumes higher technical proficiency

The "Bus Factor" Reality:

If technical volunteer leaves:

- Code to understand: ~800-1,200 lines (more complex configuration)
- Skills required: Intermediate JavaScript, Scheduler-specific knowledge
- Time to onboard: 2-3 days
- Finding replacement: Fewer developers familiar with DHTMLX vs. FullCalendar
- Commercial support: Available with paid license ($599+)

More complex than FullCalendar. Feature-rich but steeper learning curve for future volunteers.

Development Estimate

Total: 8-12 days developer time (display-only scope)

- More configuration options mean more decisions
- Less community content means more trial-and-error
- Accessibility features may require additional testing time
- Brand customization more complex than FullCalendar

Member Pain Point Resolution

Addresses same core issues as FullCalendar but:

- May conflict with “I want to do my business and get out” preference
- Feature richness could overwhelm simple use case
- Timeline/Agenda views more complex than SBNC needs

Pros for SBNC

- Solid enterprise features if complex scheduling needed later
- Strong accessibility (important for inclusive design)
- Commercial support available with paid license
- Free version exists for basic use
- Good brand customization capabilities

Cons for SBNC

- Steeper learning curve for volunteers
- Smaller community than FullCalendar
- May be over-engineered for SBNC’s needs
- Pro features require $599+ license
- More complex brand customization process

---

4. Mobiscroll Event Calendar

Implementation Approach: Premium mobile-first JavaScript calendar library (display-only)

Product Information: - Website: https://mobiscroll.com - Event Calendar: https://mobiscroll.com/javascript/event-calendar - Documentation: https://docs.mobiscroll.com/javascript/eventcalendar - Pricing: https://mobiscroll.com/pricing (starts at $499/year) - License: Commercial only

Member Requirements Coverage

Best-in-class mobile experience. Native feel on smartphones addresses the “members prefer mobile app” problem directly. Beautiful, modern aesthetic addresses “dated design” concerns.

Brand Customization

Rating: ★★★★★ (Excellent)

Capabilities:

- Sophisticated theming system designed for brand consistency
- SBNC colors can be precisely matched across all components
- Mobile-first design means brand consistency on all devices
- Premium aesthetic aligns with modern design expectations
- Full font control

Implementation:

- Theme builder tool simplifies brand color application
- CSS variables for consistent styling
- Documentation includes branding best practices
- Touch-optimized elements maintain brand feel on mobile

Maintenance & Support Profile

Core Library Health:

- Commercial product from Mobiscroll (10+ years)
- Regular updates and new features
- License: Commercial only ($499-999/year depending on features)
- Company: Acid Media (established mobile UI library provider)

Support Characteristics:

- Commercial support included with license
- Documentation clear and mobile-focused
- Community smaller than open-source options
- Direct vendor support via support tickets

The "Bus Factor" Reality:

If technical volunteer leaves:

- Code to understand: ~500-800 lines
- Skills required: Basic JavaScript, mobile-first development
- Time to onboard: 1-2 days
- Commercial support: Included with license (advantage over open-source)
- Finding replacement: Fewer developers know Mobiscroll specifically, but code patterns similar to other libraries

Commercial support included. If volunteer unavailable, can pay vendor for help.

Development Estimate

Total: 5-7 days developer time (display-only scope)

- Similar to FullCalendar (read WA API, display events, link to WA for registration)
- Mobile optimization built-in (less testing)
- Premium documentation accelerates development
- Brand theming tools reduce styling time

Member Pain Point Resolution

Member Quote	How Mobiscroll Solves It
“I mostly use the app, someone said that’s the way to do it”	Mobile-first design narrows website vs. app gap
“It feels very dated”	Modern, polished aesthetic
“Calendar navigation loses position”	Smooth single-page app behavior

Strongest solution for mobile-heavy user base. Best addresses the preference gap between mobile app and website.

Pros for SBNC

- Best mobile experience (directly addresses app vs. website gap)
- Commercial support included (reduces volunteer burden)
- Modern aesthetic addresses “dated” complaints
- Touch-optimized interactions feel native on phones
- Vendor handles library maintenance and updates
- Excellent brand customization with premium feel

Cons for SBNC

- Annual license cost ($499-999/year)
- Smaller community for troubleshooting
- License cost might limit budget for other improvements
- Still requires JavaScript developer for implementation

---

5. Schedule-X

Implementation Approach: Modern, lightweight JavaScript calendar library (display-only)

Product Information: - Website: https://schedule-x.dev - Documentation: https://schedule-x.dev/docs/calendar/introduction - GitHub: https://github.com/schedule-x/schedule-x - License: MIT (open source) - Demo: https://schedule-x.dev/demos

Member Requirements Coverage

Clean, minimal design aligns with “utility over aesthetics” preference. Lightweight and fast. Built with modern web standards.

Brand Customization

Rating: ★★★★☆ (Very Good)

Capabilities:

- Modern CSS approach with full control
- SBNC brand colors can be applied via CSS variables

- Clean, minimal default styling reduces customization needed
- Font control available
- Lightweight means faster brand asset loading

Implementation Consideration:

- Being newer means fewer branding examples in documentation
- Modern CSS practices (CSS Grid, Flexbox) required
- Less opinionated default styling gives more freedom

Maintenance & Support Profile

Core Library Health:

- Repository: github.com/schedule-x/schedule-x
- Stars: 1,200+ | Age: 2 years | Growing community
- License: MIT (open source)
- Creator: Tom Österlund (active maintainer)

Support Characteristics:

- Newer library means less community content
- Stack Overflow: ~200 questions (limited compared to others)
- GitHub issues are responsive but smaller community
- Documentation good but fewer examples than mature libraries

The “Bus Factor” Reality:

If technical volunteer leaves:

- Code to understand: ~600-1,000 lines
- Skills required: Modern JavaScript (ES6+), API integration
- Time to onboard: 2-4 days (less documentation available)
- Finding replacement: Fewer developers know Schedule-X
- Library risk: Newer library, less proven track record

Being newer means:

- Less battle-tested in production environments
- Fewer solved problems on Stack Overflow
- Higher risk of breaking changes
- Smaller community to help troubleshoot

This doesn’t mean it’s bad, but volunteer organizations benefit from mature, proven tools.

Development Estimate

Total: 6-10 days developer time (display-only scope)

- Less documentation means more experimentation

- Fewer examples means building more from scratch
- Modern patterns require up-to-date JavaScript knowledge
- Brand customization less documented

Member Pain Point Resolution

Addresses core issues but with less proven track record than FullCalendar. Lightweight design aligns with “utility” preference but implementation risk higher.

Pros for SBNC

- Free and open-source
- Modern, clean codebase
- Lightweight and fast
- Responsive design built-in
- Good for organizations comfortable with newer technology
- Full brand customization control

Cons for SBNC

- Newer library (2 years vs. 15 for FullCalendar)
- Smaller community
- Less documentation and examples
- Higher risk if library development slows
- More difficult to find developers familiar with it
- Less guidance on brand customization

Iframe Solutions: Detailed Analysis

Understanding Iframe Trade-offs

What you gain:

- Zero maintenance (vendor handles updates, security, hosting)
- No JavaScript skills required
- Quick implementation (hours, not days)
- Visual admin interface (no code changes)
- Vendor support included

What you sacrifice:

- Limited customization of appearance (including brand colors/fonts)
- Cannot deeply integrate with website design
- Responsiveness depends on vendor implementation
- Cross-origin restrictions limit interactions
- Data lives on third-party servers

6. TeamUp Calendar

Implementation Approach: Embeddable calendar service via iframe

Product Information: - Website: https://teamup.com - Features: https://www.teamup.com/features - Pricing: https://www.teamup.com/pricing (Free - $20/month) - Help Center: https://help.teamup.com - Embedding Guide: https://help.teamup.com/hc/en-us/articles/360010481494-Embed-a-calendar-in-a-website

Overview

TeamUp is designed for embedding calendars on external websites. Popular with non-profits, sports leagues, and community organizations. Provides visual admin interface for managing events without technical skills.

Member Requirements Coverage

Workshop Requirement	TeamUp Implementation	Rating
Clean, scannable calendar	Multiple view options, clean UI	★★★★★
Filter by event type	Sub-calendars with color-coding	★★★★☆
Show Open/Full/Waitlist	Custom fields + signups	★★★☆☆
Calendar + List views	Month/Week/Day/Agenda views	★★★★★
Text search	Built-in search functionality	★★★★★
Preserve position	Generally good, some iframe scroll issues	★★★★☆
Mobile responsive	Good mobile support	★★★★☆
Custom brand colors/fonts	Limited (theme selection)	★★☆☆☆

Brand Customization

Rating: ★★☆☆☆ (Fair)

Capabilities:

- Theme selection from vendor-provided options
- Limited color customization (primary color selection)
- Cannot precisely match SBNC brand colors (teal, purple, golden orange)
- No font control
- Sub-calendar colors can be customized within available palette

Limitations:

- Iframe-based means SBNC cannot apply custom CSS
- Vendor controls overall design aesthetic

- Brand consistency depends on selecting closest theme match
- Cannot apply SBNC typography guidelines

Trade-off: Sacrifice brand precision for zero maintenance burden.

Maintenance & Support Profile

Zero Maintenance for SBNC:

- TeamUp handles all updates, security patches, hosting
- No code to maintain
- No developer needed after initial setup
- Visual interface for admin tasks

The “Bus Factor” Solution:

If your technical volunteer leaves, next person needs:

1. Code to understand: None (visual admin interface)
2. Skills required: Ability to use web applications
3. Time to onboard: 1 hour to learn admin interface
4. Finding replacement: Any board member can manage

This is the lowest “bus factor” risk of all solutions.

Support:

- Help center with documentation
- Email support (response within 1 business day)
- Active user community
- Video tutorials

Integration with Wild Apricot

TeamUp is separate system from Wild Apricot. Two approaches:

Option A: TeamUp for Display Only

- Manually copy events from WA to TeamUp calendar
- Users click event in TeamUp, redirected to WA for registration
- Pro: Simple, low risk
- Con: Duplicate data entry

Option B: API Sync (Requires developer)

- Script syncs WA events to TeamUp via APIs
- Automated daily sync keeps calendars aligned
- Pro: No duplicate entry
- Con: Requires developer to build sync script (~2-3 days)
- Maintenance: Sync script needs occasional updates

Pricing

- **Free Plan:** 1 calendar, 100 events, TeamUp branding
- **Plus Plan:** $8/month, 8 sub-calendars, 1,000 events, remove branding
- **Pro Plan:** $20/month, unlimited sub-calendars and events, priority support

Recommendation: Plus Plan ($8/month = $96/year) sufficient for SBNC needs

Member Pain Point Resolution

Member Quote	How TeamUp Addresses It
"You click something and go back, you're back at the top"	Better position handling than WA native
"The app has a search bar, but website doesn't"	Built-in search functionality
"Helpful to know which events still have openings"	Custom fields can show capacity info
"I do find the calendaring part clunky"	Modern, smooth interface

Limitation: Still requires click-through to Wild Apricot for actual registration. Two-system flow.

Pros for SBNC

- Zero maintenance burden
- No technical skills required
- Visual admin interface (any board member can use)
- Built-in search and filtering
- Vendor handles updates and security
- Very low "bus factor" risk
- Month-to-month pricing ($8/mo, can cancel anytime)

Cons for SBNC

- Separate system from Wild Apricot (duplicate data or sync script needed)
- Users click through from TeamUp to WA for registration (two-step process)
- Limited design customization (cannot match SBNC brand precisely)
- Annual cost ($96-240/year)
- Data hosted by third party
- Cannot show real-time WA registration status without sync script
- Limited brand color/font control

---

7. Google Calendar Public Embed

Implementation Approach: Embed Google Calendar iframe on website

Product Information: - Website: https://calendar.google.com - Google Workspace:

https://workspace.google.com/products/calendar/ - Embedding Guide: https://support.google.com/calendar/answer/41207 - API Documentation: https://developers.google.com/calendar

Overview

Create public Google Calendar, manually add events, embed using Google's iframe code. Most organizations already use Google Workspace, making this familiar territory.

Member Requirements Coverage

Workshop Requirement	Google Calendar	Rating
Clean, scannable calendar	Standard Google Calendar interface	★★★☆☆
Filter by event type	Multiple calendars with layer toggle	★★☆☆☆
Show Open/Full/Waitlist	Not supported (description text only)	★☆☆☆☆
Calendar + List views	Month/Week/Schedule views	★★★☆☆
Text search	Basic search in calendar	★★☆☆☆
Preserve position	Variable (iframe scrolling issues)	★★☆☆☆
Mobile responsive	Better on Google's site than embedded	★★★☆☆
Custom brand colors/fonts	Minimal (basic colors only)	★★☆☆☆

Brand Customization

Rating: ★★☆☆☆ (Fair)

Capabilities:

- Can set one primary color for calendar
- Cannot match multiple SBNC brand colors (teal, purple, orange)
- No font control
- Google Calendar's visual design is fixed
- Event colors can be set but limited palette

Limitations:

- Iframe-based prevents custom CSS
- Cannot apply SBNC typography
- Visual identity remains clearly "Google Calendar"
- Limited ability to create cohesive brand experience

Trade-off: Familiar Google interface but minimal brand alignment.

Integration with Wild Apricot

Google Calendar is completely separate from Wild Apricot.

Manual Approach:

- Admin copies events from WA to Google Calendar
- Update event descriptions with WA registration links
- Extremely time-consuming, error-prone

Automated Approach:

- Script syncs WA events to Google Calendar via APIs
- Requires developer to build (~2-3 days)
- Make.com or Zapier can automate (~1 day setup, $15-30/mo)

Maintenance & Support Profile

Minimal Maintenance:

- Google handles platform updates
- No custom code unless using automated sync
- Any admin can manage Google Calendar

The "Bus Factor":

- Zero risk for calendar platform itself
- If using sync automation, need to maintain that integration

Pricing

- Free if organization has Google Workspace
- Public calendar is free even without Workspace
- Automation: $15-30/month if using Zapier/Make.com

Member Pain Point Resolution

Member Quote	Google Calendar Performance
"You click something and go back, you're back at the top"	Still occurs (iframe limitation)
"The app has a search bar, but website doesn't"	Basic search available
"Helpful to know which events still have openings"	Cannot show dynamically
"I do find the calendaring part clunky"	Google UI familiar but basic

Does not strongly address top pain points.

Pros for SBNC

- Free (if already using Google Workspace)

- Familiar interface (everyone knows Google Calendar)
- Zero maintenance for calendar platform
- Easy for any admin to manage
- Can set up in 1-2 hours

Cons for SBNC

- Separate system from Wild Apricot (duplicate data)
- Cannot show registration status dynamically
- Limited customization (minimal brand control)
- Embed has iframe scroll issues
- Better experience going directly to Google Calendar than embedding
- Two-step process (view event in Google, register in WA)
- Doesn’t strongly address workshop pain points
- Cannot match SBNC brand colors/fonts

8. Calendly / AddEvent Style Solutions

Implementation Approach: Booking/event widget services

Product Information: - Calendly: https://calendly.com - AddEvent: https://www.addevent.com - Skedda: https://www.skedda.com

Overview

Services like Calendly (scheduling), AddEvent (event widgets), Skedda (booking) are designed for different use cases than SBNC needs:

- **Calendly:** One-on-one appointment scheduling
- **AddEvent:** “Add to calendar” buttons
- **Skedda:** Facility/resource booking

Assessment

These solutions are designed for different use cases than SBNC requires:

- **Calendly:** One-on-one appointment scheduling
- **AddEvent:** “Add to calendar” buttons for generating ICS files
- **Skedda:** Facility/resource booking

SBNC requires event listing and registration management, which differs from these tools' primary purposes.

Brand Customization

Rating: ★☆☆☆☆ (Poor)

Capabilities:

- Minimal customization available
- Tools are designed for their specific use cases, not general event calendars
- Cannot match SBNC brand guidelines
- Limited to vendor’s design aesthetic

These tools are not designed for SBNC’s use case.

Side-by-Side Technical Comparison

Browser & Device Compatibility

Solution	Desktop Chrome/Edge	Desktop Safari	Mobile Chrome	Mobile Safari	Tablet	Notes
WA Native	Fair	Fair	Poor	Poor	Fair	Known Safari issues
FullCalendar	Excellent	Excellent	Excellent	Excellent	Excellent	Thoroughly tested
DHTMLX	Excellent	Good	Good	Good	Good	Some mobile tweaking may be needed
Mobiscroll	Excellent	Excellent	Excellent	Excellent	Excellent	Mobile-first design
Schedule-X	Good	Good	Good	Good	Good	Modern browsers only
TeamUp	Good	Good	Fair	Fair	Fair	Iframe scroll issues possible
Google Calendar	Good	Good	Fair	Fair	Fair	Better experience outside iframe

Performance Characteristics

Solution	Initial Load	Event Render	Large Dataset (500+ events)	Mobile Performance
WA Native	Slow (iframe + WA overhead)	Moderate	Poor	Slow
FullCalendar	Fast	Fast	Good (pagination needed)	Fast
DHTMLX	Moderate	Fast	Excellent	Moderate
Mobiscroll	Fast	Fast	Good	Excellent
Schedule-X	Very Fast	Fast	Good	Fast
TeamUp	Moderate (iframe)	Fast	Good (vendor-managed)	Moderate
Google Calendar	Fast	Fast	Excellent	Fast

Integration Complexity

Solution	WA API Integration	Authentication	Data Sync	Complexity Level
Current Solution				
WA Native	Built-in	Built-in	Real-time	Very Low
JavaScript Solutions				
FullCalendar	Direct API calls	Token-based	Real-time pull	Moderate
DHTMLX	Direct API calls	Token-based	Real-time pull	Moderate-High
Mobiscroll	Direct API calls	Token-based	Real-time pull	Moderate
Schedule-X	Direct API calls	Token-based	Real-time pull	Moderate
Iframe Solutions				
TeamUp	Manual or script sync	N/A or script-managed	Periodic push	Low (manual) / Moderate (automated)
Google Calendar	Script sync required	OAuth2	Periodic push	Moderate

Maintainability Deep Dive: The Volunteer Organization Reality

The Core Challenge

SBNC faces a reality common to volunteer organizations:

- **High Turnover:** Board members rotate, typically 1-2 year terms
- **Variable Skills:** Next technology chair may have different skill level
- **Time Constraints:** Volunteers have limited hours
- **No Institutional Memory:** When volunteer leaves, knowledge leaves
- **Budget Limitations:** Cannot afford permanent technical staff

This means the calendar solution must account for:

- **Knowledge Transfer:** How easily can next volunteer understand the implementation?
- **Emergency Support:** What happens if something breaks and no one knows how to fix it?
- **Update Burden:** Who keeps integrations working when APIs change?
- **Documentation Decay:** Will future volunteers understand past decisions?

Maintenance Burden by Solution Type

JavaScript Solutions: Moderate to High Burden

Ongoing Maintenance Tasks:

1. Wild Apricot API Changes: When WA updates their API, integration code needs updates
2. Library Updates: Keep calendar library updated for security and features
3. Browser Changes: Keep compatible with browser updates
4. Bug Fixes: Debug issues when they arise
5. Feature Requests: Implement new filtering or display requests

Required Skills:

- JavaScript programming
- Understanding of APIs
- Debugging ability
- Source control (Git)

Time Commitment:

- Initial build: 5-10 days
- Ongoing maintenance: ~4-8 hours per quarter

Bus Factor Risk: Moderate to High

- If technical volunteer leaves, need to find replacement with JavaScript skills
- Could take weeks or months to find suitable volunteer
- In emergency, can hire developer at $50-150/hour

Mitigation Strategies:

1. **Thorough Documentation**
    - Code comments explaining every major function
    - Separate README with architecture decisions
    - Step-by-step guide for common changes
2. **Modular Architecture**
    - Separate WA integration from calendar display
    - Configuration in JSON files (non-developers can edit)
    - Clear file structure
3. **Budget Reserve**
    - Set aside $500-1000 for emergency developer help
    - Identify freelancers in advance
4. **Backup Plan**
    - Know you can revert to WA native if necessary
    - Keep sync scripts even if not actively used

Iframe Solutions: Low to Zero Burden

Ongoing Maintenance Tasks:

1. Event Entry: Manual data entry or maintaining sync scripts
2. Settings Changes: Occasional updates via web interface
3. Monitoring: Check that embed is displaying correctly

Required Skills:

- Using web applications
- Basic troubleshooting

Time Commitment:

- Initial setup: 1-2 hours
- Ongoing maintenance: ~1 hour per month (if manual entry)

Bus Factor Risk: Very Low

- Anyone can manage web-based admin interface
- Vendor handles all technical updates
- Support available from vendor

Trade-off:

- Low maintenance burden but limited customization (including brand colors/fonts)
- May not address all member pain points

- Dependent on third-party service

Maintenance Cost Comparison (3-Year Total Cost of Ownership)

Solution	Year 1	Year 2	Year 3	3-Year Total	Notes
WA Native					
All Costs	$0	$0	$0	$0	Included in WA subscription
Member Frustration	???	???	???	???	Unquantifiable cost
FullCalendar.io					
Development	$2,000	-	-	$2,000	5 days @ $400/day (volunteer time value)
Maintenance	$400	$400	$400	$1,200	1 day/year @ $400 (volunteer time value)
Emergency Support	$300	-	$300	$600	2 incidents @ $300
Subtotal	**$2,700**	**$400**	**$700**	**$3,800**	
Mobiscroll					
License	$799	$799	$799	$2,397	Annual license
Development	$2,000	-	-	$2,000	5 days @ $400/day
Maintenance	-	-	-	-	Vendor support included
Subtotal	**$2,799**	**$799**	**$799**	**$4,397**	
TeamUp Calendar					
Subscription	$96	$96	$96	$288	Plus plan @ $8/month
Setup	$200	-	-	$200	4 hours @ $50/hr (volunteer time value)
Maintenance	$600	$600	$600	$1,800	12 hours/year manual entry @ $50/hr
Subtotal	**$896**	**$696**	**$696**	**$2,288**	

When accounting for volunteer time value, TeamUp shows the lowest 3-year total cost of ownership. FullCalendar has higher upfront investment but no recurring license fees. WA Native has zero implementation and maintenance costs but does not address workshop-documented member pain points.

Documentation Requirements by Solution

JavaScript Solutions: Full Documentation Needed

Must Document:

1. **Architecture Overview**
   - How calendar connects to WA API (read-only, display-only)
   - Where event data is cached
   - How filters work
2. **Code Map**
   - File structure
   - What each file does
   - Where to find key functions
3. **Common Changes**
   - How to add new event type filter
   - How to change colors (brand updates)
   - How to update WA API credentials
4. **Troubleshooting Guide**
   - Common error messages
   - How to check if WA API is working
   - Browser console debugging basics
5. **Deployment Process**
   - How to push changes live
   - How to roll back if needed
6. **Contact Information**
   - Original developer contact
   - Backup developers identified

Estimated Documentation Time: 1-2 days (critical investment)

Iframe Solutions: Minimal Documentation Needed

Must Document:

1. **Login Credentials**
   - Where to access admin interface
   - Password reset process
2. **Common Tasks**
   - How to add an event
   - How to update settings
   - How to embed on different pages
3. **Contact Information**

    - Vendor support contact

Estimated Documentation Time: 2-3 hours

Knowledge Transfer Scenarios

Scenario 1: Planned Handoff (New Technology Chair)

JavaScript Solution:

- Outgoing volunteer schedules 2-3 hour session
- Walk through code and documentation
- New volunteer makes supervised change
- Review common troubleshooting issues
- Share developer contacts

Iframe Solution:

- 30-minute walk-through of admin interface
- New volunteer adds test event
- Share login credentials

Scenario 2: Emergency (Volunteer Suddenly Unavailable)

JavaScript Solution:

- Review documentation (hopefully exists and is current)
- Attempt to understand code
- If cannot fix, contact backup developers
- If urgent, hire freelancer ($150-500 depending on issue)
- Consider reverting to WA native temporarily

Iframe Solution:

- Log into vendor admin interface
- Contact vendor support if issue unclear
- No code to understand or debug

Scenario 3: API Breaking Change (Wild Apricot Updates API)

JavaScript Solution:

- WA announces API v3 deprecated, must upgrade to v4
- Review WA migration documentation
- Update integration code
- Test thoroughly
- Requires JavaScript skills
- 4-8 hours work

Iframe Solution with Sync:

- Update sync script (if using automation)
- 2-4 hours work

Iframe Solution without Sync:

- No impact (manual entry continues)

Support Network Comparison

FullCalendar.io Support Network

Free Resources:

- Official documentation (thorough)
- Stack Overflow (7,500+ questions)
- GitHub issues (active)
- Community forum
- Hundreds of blog tutorials
- YouTube videos

Paid Resources:

- FullCalendar LLC support: $200-500 per incident
- Freelance developers: $50-150/hour
- Consulting firms: $150-250/hour

Local Resources:

- Many Santa Barbara web developers know FullCalendar
- SB Tech community could provide recommendations

TeamUp Support Network

Free Resources:

- Help center documentation
- Video tutorials
- User community forum

Paid Resources:

- Included: Email support (1 business day response)
- Included: Priority support with Pro plan
- Very limited freelance market (proprietary platform)

Local Resources:

- No local expertise needed (web interface)

The 2 AM Test

Imagine: Friday night, 2 AM, board member wakes you up because the calendar is broken and there’s a major event signup opening Saturday morning.

JavaScript Solution:

- Can you (or successor) debug JavaScript?
- Is documentation good enough to understand the code?
- Can you access source control to see recent changes?
- Do you have backup developer contacts?
- Can you roll back to previous version?
- Can you hire emergency help if needed?

Iframe Solution:

- Log into vendor admin interface
- Check recent settings changes
- Contact vendor support (email, response next business day)
- Worst case: Temporarily link directly to WA calendar

WA Native:

- It already doesn’t work well, so member expectations are low
- Contact WA support
- No options to fix core issues

Each scenario presents different capabilities and constraints for handling emergency situations. JavaScript solutions require technical troubleshooting capabilities, while iframe solutions rely on vendor response times. Current WA Native solution has limited emergency response options due to platform limitations.

Implementation Roadmap

Phase 1: MVP (Target: January 15, 2025)

Timeline: 6 weeks from December 1 start date

Week 1-2: Solution Selection & Preparation

Days 1-3: Decision Making

- Review this analysis with technology committee
- Validate budget availability
- Confirm technical volunteer availability (if choosing JavaScript)
- Make go/no-go decision

Days 4-5: Preparation

- If JavaScript: Set up development environment
- If iframe: Select vendor and create account

- Document current WA calendar configuration
- Create test event dataset

Days 6-10: Planning

- Define exact event type categories for filtering
- Design color scheme for status indicators (using SBNC brand colors)
- Map WA event fields to calendar display
- Write basic documentation structure
- Define brand application strategy (colors, fonts, styling)

Week 3-4: Core Implementation

JavaScript Solution Path:

- Days 11-12: Basic calendar display with WA API (read-only, display-only)
- Days 13-14: Event type filtering
- Day 15: Status indicators (Open/Full/Waitlist)
- Day 16: Text search
- Days 17-18: Brand customization (SBNC colors/fonts) and mobile responsiveness testing
- Days 19-20: Bug fixing and refinement

Iframe Solution Path (TeamUp):

- Day 11: TeamUp account setup and configuration
- Day 12: Define sub-calendars for event types, configure colors
- Day 13: Manual event entry (first 10-20 events)
- Day 14: Embedding and iframe configuration
- Day 15: Mobile testing
- Days 16-20: Complete event entry, testing

Week 5: Testing & Documentation

Days 21-22: Internal Testing

- Technology committee members test all features
- Board members test on various devices
- Document bugs and issues

Days 23-24: User Acceptance Testing

- Recruit 5-10 members for testing
- Observe them using calendar
- Collect feedback

Day 25: Documentation

- Write user guide for members
- Complete admin documentation
- Document known issues and workarounds

Week 6: Refinement & Launch

Days 26-28: Bug Fixes

- Address critical issues from testing
- Refine based on user feedback
- Performance optimization

Days 29-30: Soft Launch

- Replace WA native calendar with new solution
- Announce to board first
- Monitor for issues

Days 31-35: Member Communication

- Email to members explaining improvements
- Post on website
- Monitor member feedback

Day 35: Workshop Deadline (January 15)

- Solution live and stable
- Collecting member feedback
- Celebrating success

Phase 2: Enhancements (January 16 - March 1)

Nice-to-Have Features:

- Additional filter tags (couples-friendly, solo-friendly, family-friendly)
- “My Events” dashboard integration
- Show member’s registration status on event page
- Advanced search (by location, date range, price range)
- Performance optimizations
- Brand refinements based on member feedback

Timeline: Implement 1-2 enhancements per month based on:

- Member feedback from Phase 1
- Technical volunteer availability
- Budget constraints

Phase 3: Measurement & Iteration (March 1+)

Success Metrics (from Workshop):

- Reduced clicks to register for events
- Increased website usage vs. app
- Faster event discovery time

- Fewer support requests
- Improved perception of modernity

Measurement Methods:

- Google Analytics: Track calendar page views, time on page
- Survey: “How would you rate the new calendar?”
- Support tickets: Compare calendar-related inquiries before/after
- Board feedback: Qualitative input from event chairs

Quarterly Reviews:

- Review metrics with technology committee
- Identify pain points
- Prioritize next enhancements
- Budget for ongoing maintenance

Risk Assessment

Implementation Risks

Risk	Probability	Impact	Mitigation
Developer unavailable before deadline	Medium	High	Start immediately; identify backup developer; consider iframe solution
Technical complexity higher than estimated	Medium	Medium	Use mature library (FullCalendar); allow buffer time
WA API limitations discovered	Low	High	Research API thoroughly during planning; have backup plan
Member adoption slow	Low	Medium	Clear communication; user guide; board champions
Browser compatibility issues	Low	Medium	Test early and often on all major browsers
Budget constraints emerge	Low	High	Start with free solution; defer paid features
Brand customization more complex than expected	Low	Medium	Allow extra time in schedule; use solutions with good theming documentation

Operational Risks

Risk	Probability	Impact	Mitigation
Technical volunteer leaves after launch	High	High	Full documentation; identify backup resources; consider iframe solution
WA updates break integration	Medium	Medium	Subscribe to WA API announcements; test updates in staging; maintain relationships with WA support
Member confusion with new interface	Medium	Low	User guide; gradual rollout; collect feedback early
Performance issues with large datasets	Low	Medium	Implement pagination; cache event data; optimize API calls
Vendor service discontinuation (iframe)	Low	High	Choose established vendors; maintain ability to export data; have backup plan

Long-Term Sustainability Risks

Risk	Probability	Impact	Mitigation
Knowledge loss through volunteer turnover	High	High	Full documentation; multiple people with knowledge; regular training
Code becomes outdated/unmaintainable	Medium	High	Choose widely-used library; keep dependencies updated; periodic code reviews
Budget cannot support paid solution	Low	Medium	Start with free/low-cost; demonstrate ROI; build into annual budget
Organization priorities shift	Low	Medium	Align with strategic goals; demonstrate member value; board buy-in
Brand guidelines change	Low	Low	Use CSS variables for easy updates; document brand application process

Summary of Analysis

This analysis evaluated eight calendar solutions across two implementation approaches:

JavaScript Solutions: Custom development integrating Wild Apricot API with calendar libraries for display purposes only. Provides maximum control over features, functionality, and brand customization. Requires technical resources for implementation and ongoing maintenance.

Iframe Solutions: Third-party services embedded via iframe. Minimal technical requirements. Vendor manages hosting, updates, and maintenance. Limited customization capability, particularly for brand alignment.

Key Trade-offs Identified

Feature Capability vs. Maintenance Burden:

- JavaScript solutions offer superior feature sets and complete brand control but require ongoing technical volunteer availability
- Iframe solutions provide simpler maintenance at the cost of customization depth and brand precision

Brand Customization:

- JavaScript solutions (FullCalendar, DHTMLX, Mobiscroll, Schedule-X): Complete CSS control for exact matching of SBNC colors (teal, purple, golden orange) and typography
- WA Native: Good customization via Colors & Styles and CSS, though limited by platform constraints
- Iframe solutions (TeamUp, Google Calendar): Limited to vendor themes; cannot precisely match brand guidelines

Cost Structures:

- Free open-source options (FullCalendar, Schedule-X) have no license fees but require development investment
- Commercial options (Mobiscroll, TeamUp) have ongoing costs but include vendor support
- Current WA native solution is included in existing subscription but does not address documented member pain points

Risk Profiles:

- JavaScript solutions carry “bus factor” risk related to volunteer turnover and technical knowledge transfer
- Iframe solutions minimize this risk through vendor-managed platforms and visual admin interfaces
- All solutions except WA native require some initial implementation effort

Solutions Overview

Current Solution:

- **Wild Apricot Native:** Included in subscription, does not address workshop pain points, good brand customization possible via CSS

High Feature Capability, Moderate Maintenance:

- **FullCalendar.io:** Most mature (15 years), largest community, free, excellent brand customization
- **DHTMLX Scheduler:** Enterprise features, accessibility focus, free/$599+, very good brand customization
- **Mobiscroll:** Mobile-first, commercial support, $499-999/year, excellent brand customization

Moderate Feature Capability, Lower Maintenance:

- **Schedule-X:** Modern/lightweight, newer (2 years), free, good brand customization
- **TeamUp Calendar:** Purpose-built for embedding, $8-20/month, zero code maintenance, limited brand customization

Basic Capability, Minimal Maintenance:

- **Google Calendar:** Free, familiar interface, limited integration, minimal brand customization

Implementation Time Ranges

- **Current WA Native:** 0 days (already implemented), potential CSS customization: 1-2 days
- **JavaScript solutions:** 4-11 days depending on complexity (display-only scope)
- **Iframe solutions:** 0.5-2 days depending on manual vs. automated sync

Annual Cost Ranges

- **Free options:** $0 (FullCalendar, Schedule-X, Google Calendar, WA Native)
- **Low-cost:** $96-240 (TeamUp)
- **Mid-range:** $500-1000 (Mobiscroll)
- **Variable:** $0-600 (DHTMLX depending on feature needs)

Next Steps: Evaluation Process

Typical Decision-Making Timeline

Week 1: Assessment & Decision

Evaluation team typically addresses:

Solution Comparison (30-45 minutes)

- Review technical specifications
- Discuss trade-offs across solutions
- Clarify any technical questions

Organizational Capacity Assessment (20-30 minutes)

- Technical volunteer availability and timeline
- Time commitment feasibility
- Budget constraints and approval process
- Risk tolerance factors

Requirements Validation (15-20 minutes)

- Confirm workshop findings remain current
- Consider any changes since November 18
- Verify priority alignment
- Assess brand customization importance

Selection Process (15-20 minutes)

- Evaluate options against organizational context
- Identify primary solution and potential alternatives
- Define success criteria

Implementation Planning (15-20 minutes)

- Assign roles and responsibilities

- Establish communication protocols
- Set key milestone dates

Week 2: Preparation & Planning

For JavaScript Solutions:

- Development environment configuration
- Wild Apricot API documentation review (read-only access for display)
- Project planning and timeline development
- Stakeholder communication schedule
- Brand guidelines review and application planning

For Iframe Solutions:

- Vendor account creation
- Platform configuration
- Event management workflow design
- Embedding strategy development
- Brand customization within vendor constraints

Weeks 3-4: Implementation Phase

Execution follows the Phase 1 implementation roadmap detailed in the Implementation Roadmap section.

Weeks 5-6: Testing & Deployment

Testing protocols, documentation completion, and controlled rollout procedures.

January 15: Target Completion

New calendar system operational, fulfilling workshop commitment to membership.

Appendix: Technical Specifications

FullCalendar.io Implementation Details

Scope: Display-only calendar (reads WA event data, links to WA for registration)

Technology Stack:

- FullCalendar v6.x (latest)
- Vanilla JavaScript (no framework required)
- Wild Apricot API v2 (read-only access to events endpoint)
- Modern browser APIs (fetch, Promise)

File Structure:

```
/calendar/
  /js/
    fullcalendar.min.js         (library)
    wa-api.js                   (Wild Apricot API integration - read-only)
    calendar-config.js          (SBNC-specific settings)
    event-filters.js            (filtering logic)
    calendar-init.js            (initialization)
  /css/
    fullcalendar.min.css        (library styles)
    sbnc-calendar.css           (SBNC customizations)
  /config/
    event-types.json            (filterable categories)
    api-credentials.json        (WA API token - read-only permissions)
  index.html                    (calendar page)
  README.md                     (implementation documentation)
```

Wild Apricot API Integration (Read-Only):

```
// Simplified example - display only, no registration handling
async function loadEvents() {
  const response = await fetch('https://api.wildapricot.org/v2.2/accounts/{accountId}/events',
    headers: {
      'Authorization': 'Bearer ' + apiToken,  // Read-only token
      'Accept': 'application/json'
    }
  });
  const data = await response.json();

  // Format events with link to WA registration page
  return data.Events.map(event => ({
    title: event.Name,
    start: event.StartDate,
    url: `https://sbnewcomers.wildapricot.org/event-${event.Id}`,  // Link to WA
    backgroundColor: getEventColor(event),
    extendedProps: {
      status: getEventStatus(event),
      capacity: `${event.ConfirmedRegistrationsCount}/${event.RegistrationLimit || '∞'}`
    }
  }));
}
```

SBNC Brand Color Application:

```
// SBNC Brand Colors Configuration
const eventTypes = [
  { id: 'tennis', label: 'Tennis', color: '#4ECDC4' },  // SBNC Teal
  { id: 'wine', label: 'Wine Tasting', color: '#7B2CBF' },  // SBNC Purple
  { id: 'hiking', label: 'Hiking', color: '#52B788' },
  { id: 'social', label: 'Social Events', color: '#FF6B6B' },
  { id: 'dining', label: 'Dining Out', color: '#FFB627' }  // SBNC Golden Orange
];
```

Status Indicator Logic:

```
function getEventStatus(event) {
  if (event.RegistrationEnabled) {
    const registered = event.ConfirmedRegistrationsCount;
    const limit = event.RegistrationLimit;

    if (limit && registered >= limit) {
      return { status: 'full', color: '#FF6B6B', label: 'Full - Waitlist' };
    } else if (limit && registered >= limit * 0.9) {
      return { status: 'filling', color: '#FFB627', label: 'Filling Fast' };
    } else {
      return { status: 'open', color: '#52B788', label: 'Open' };
    }
  }
  return { status: 'closed', color: '#94A3B8', label: 'Registration Closed' };
}
```

TeamUp Implementation Details

Setup Process:

1. Create TeamUp account (teamup.com)
2. Create calendar with sub-calendars for event types:
    - Tennis
    - Wine Tasting
    - Hiking
    - Social Events
    - Dining Out
    - Arts & Culture
    - Learning & Classes
    - Volunteering
3. Configure calendar settings:
    - Enable search
    - Set default view (Month)
    - Enable mobile optimization
    - Configure permissions (view-only for public)
4. **Brand Customization (Limited):**
    - Select theme closest to SBNC aesthetic
    - Configure primary color (choose closest to SBNC teal)
    - Set sub-calendar colors for event types
    - Note: Cannot precisely match all SBNC brand colors

Embed code:

```
<iframe
  src="https://teamup.com/ks1234567890abcdef?view=m&showLogo=0&showSearch=1&showProfileAndInfo=
  width="100%"
  height="800px"
  frameborder="0"
  style="border:0; max-width:100%;">
</iframe>
```

Event Entry Workflow:

- Manually copy from Wild Apricot weekly
- Include WA registration link in event description
- Use custom fields for capacity information
- Color-code by sub-calendar

Automation Option (requires developer):

- Make.com or Zapier scenario
- Trigger: New event in Wild Apricot
- Action: Create event in TeamUp
- Frequency: Every 6 hours
- Cost: $15-30/month

Conclusion

The November 18 workshop identified the calendar as a CRITICAL priority based on member research documenting specific pain points. The current Wild Apricot native gadget has some platform limitations preventing resolution of several top-priority issues identified in the research, though CSS customization capabilities exist that may not be fully utilized.

Solution Pathways

Three implementation approaches have been analyzed:

Current Solution: Wild Apricot native calendar gadget. No implementation effort required. Good brand customization possible via CSS. Does not address several workshop-documented pain points due to platform limitations (position loss, no search, limited interactive filtering).

JavaScript Integration: Custom development connecting Wild Apricot API to calendar libraries for display purposes only. Offers maximum feature capability, customization, and complete brand control. Requires technical resources for implementation (4-11 days) and ongoing maintenance. Free to $1,000/year depending on library choice.

Iframe Embedding: Third-party calendar services embedded on website. Minimal technical requirements (0.5-2 days setup). Vendor handles maintenance and updates. Limited customization and brand control. $0-$240/year depending on service.

Key Considerations

The evaluation team's decision will likely weigh:

Technical Capacity:

- Availability of JavaScript developers
- Volunteer time constraints
- Long-term maintenance capabilities
- Knowledge transfer planning

Budget Parameters:

- Available funds for implementation
- Tolerance for ongoing costs
- Value assessment of features vs. price

Risk Tolerance:

- "Bus factor" concerns around volunteer turnover
- Vendor dependency considerations
- Member experience improvement urgency

Feature Priorities:

- Which workshop pain points must be addressed
- Acceptable trade-offs between capability and simplicity
- Integration depth requirements with Wild Apricot

Brand Consistency:

- Importance of precise brand color/font matching
- Acceptable level of design control
- Visual identity priorities

The analysis provides factual comparison data across these dimensions to support the evaluation team's decision-making process. The January 15 implementation deadline is achievable with most solutions given immediate decision-making and resource allocation.

Document Version: 3.0
Last Updated: November 21, 2025
Prepared For: SBNC Website Evaluation Team
Workshop Reference: November 18, 2025
Implementation Target: January 15, 2025