

Making Formal Property Verification Mainstream: An Intel® Graphics Experience

M Achutha KiranKumar V

Erik Seligman
Aarti Gupta

Bindumadhava S S
Abhijith A Bharadwaj

AGENDA

Why?

- Need for FV Deployment Drive

How?

- VPG FV Deployment Planning and Execution

What was
Achieved?

- Results of FV Deployment

Need of FV Deployment Drive

- Formal needs no introduction:
 - Established 100% state space coverage
 - NO/Minimal Bug Escapes
 - High Verification Confidence
 - Testbench Free Verification
 - Various dedicated Apps for Specific problems:
 - RTL Equivalence Sequential Check, Connectivity, X verification, Security, Low power, etc

Still...

Why are the FV adoption rates are still low?

Why was the dedicated FV deployment drive in VPG needed?

Need of FV Deployment Drive

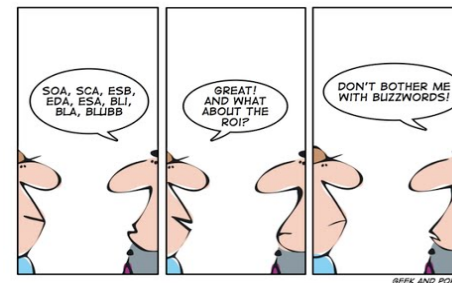
- Common FV Roadblocks:
 - FEAR:
 - MYTH: FV is only meant for PhDs
 - MYTH: Will drain my time/resources



Is it supposed to have ZERO gradient ?

- ROI:
 - TRUTH: What eventually matters?:
ROI
 - How to compare with DV results?

- No Progress Metrics
 - No intermediate check-points



AGENDA

Why?

- Need for FV Deployment Drive

How?

- VPG FV Deployment Planning and Execution

What was
Achieved?

- Results of FV Deployment

VPG FV Deployment Drive



Preparation phase
1. Survey
2. GUI Prep
3. Cookie cutter lib



Training phase
1. Champion id
2. Trainings
3. Dojos



Test planning
1. Formal TP
2. Continuous progress plan



Execution & Closure
1. Execution & Tracking
2. Bug hunting/clearance
3. Coverage metrics/closure

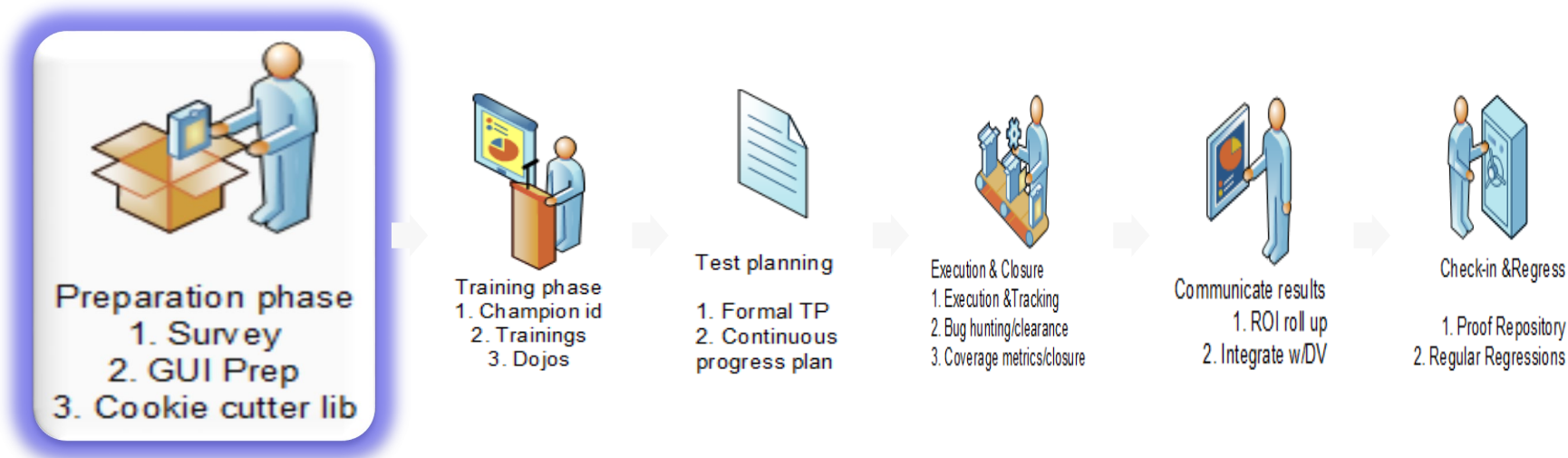


Communicate results
1. ROI roll up
2. Integrate w/DV



Check-in & Regress
1. Proof Repository
2. Regular Regressions

VPG FV Deployment Drive



VPG FV Deployment Drive:

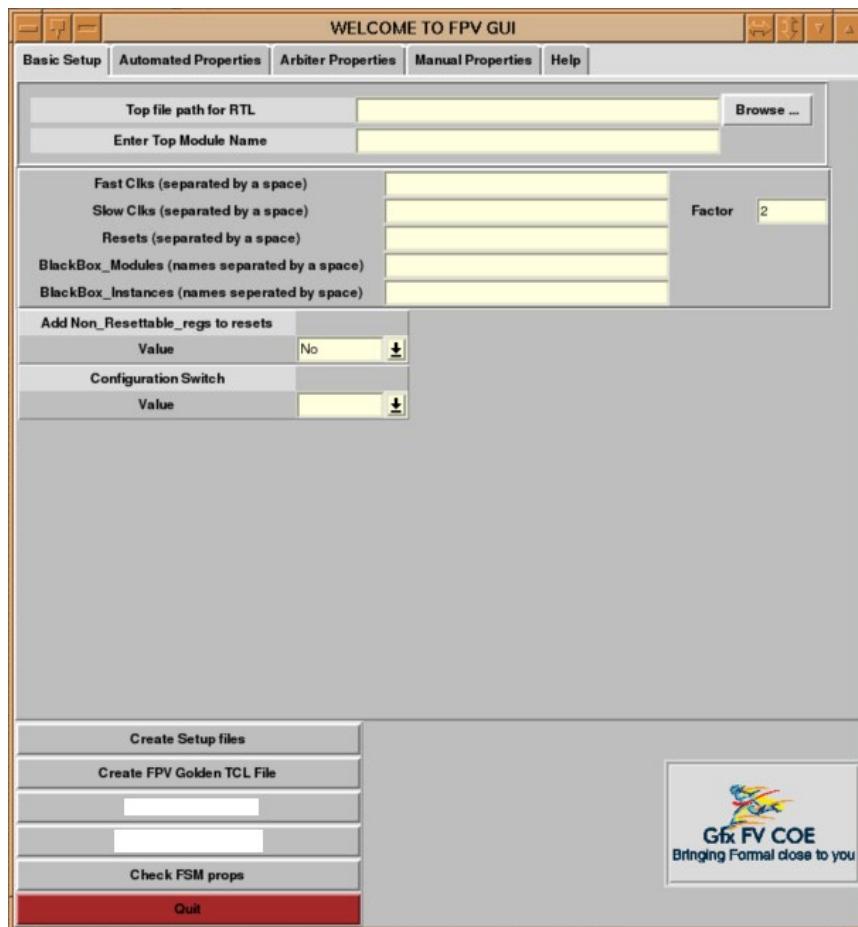
Preparation Phase

- **Survey**
 - Immediate needs for most of the users:
 - Automated setup/property_coding
 - Assertion Libraries for common design components
 - Dedicated Trainings
 - Local Contact Points



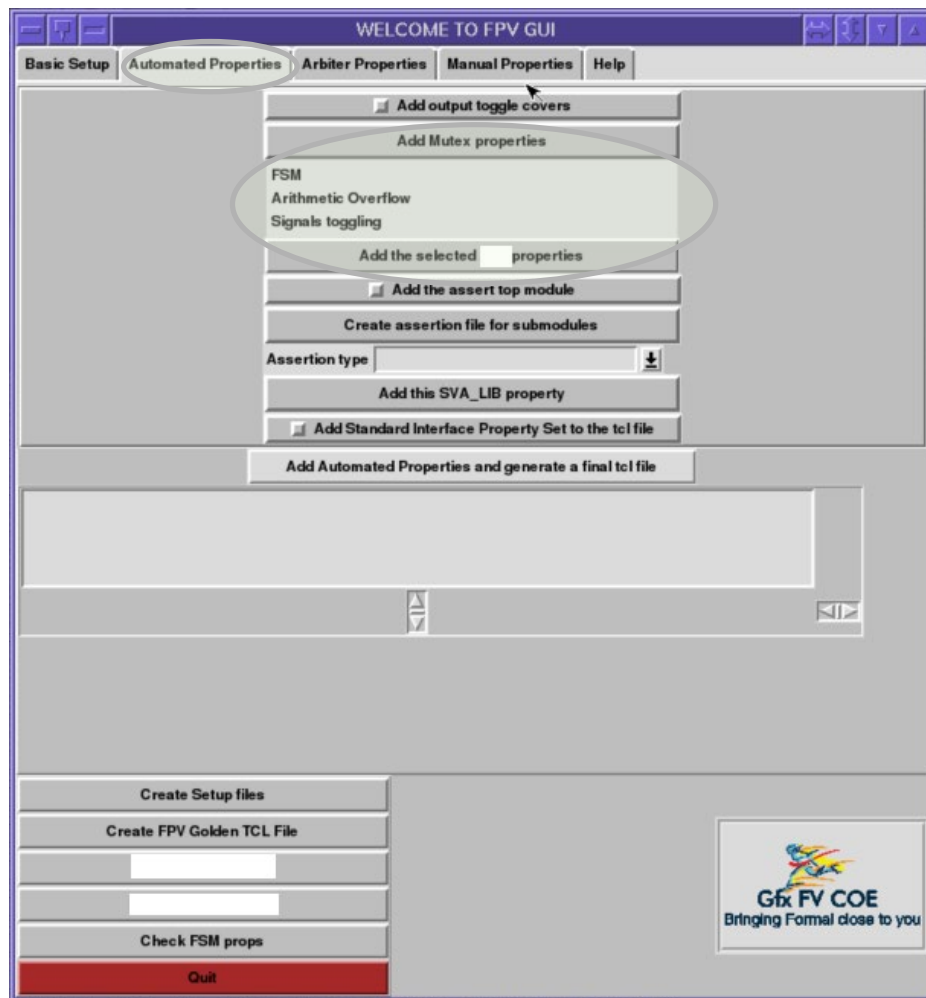
VPG FV Deployment Drive: Preparation Phase

- **GUI creation**
 - Provided a quick setup interface to new users
 - One stop shop for all formal flows
 - Easy way of generating all collaterals for formal



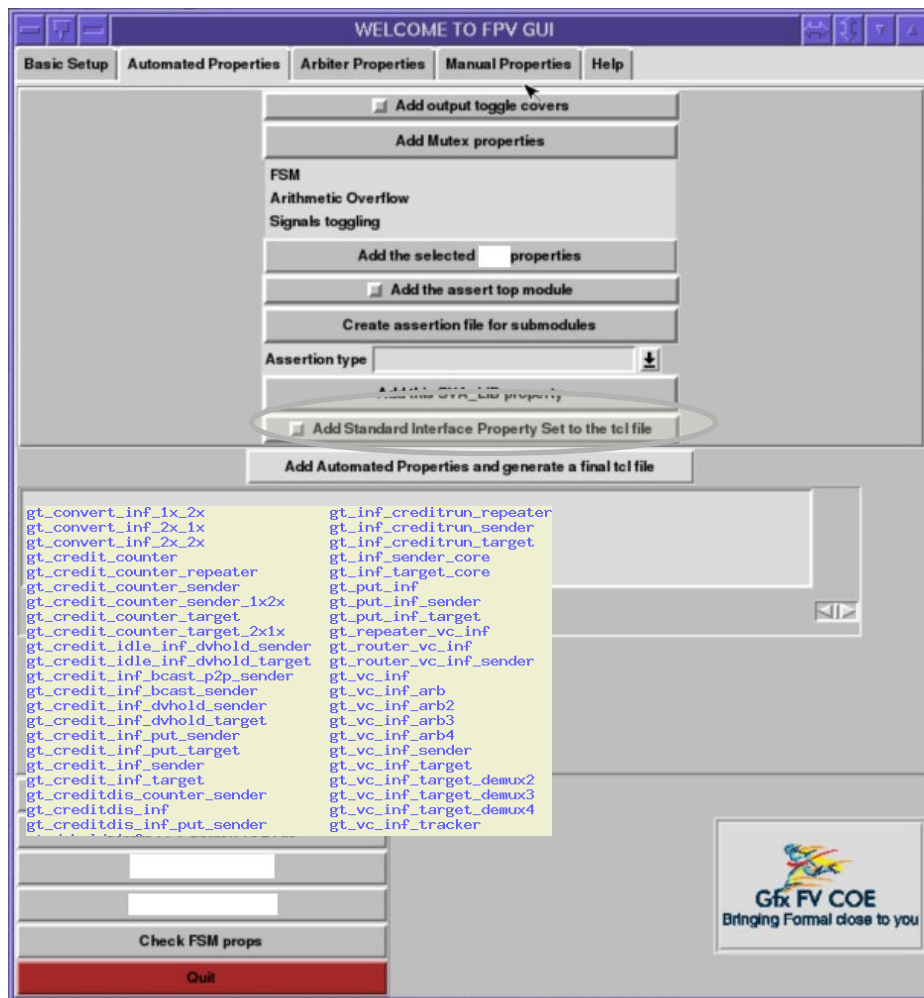
VPG FV Deployment Drive: Preparation Phase

- **Cookie-cutter Property Templates**
 - Automated properties



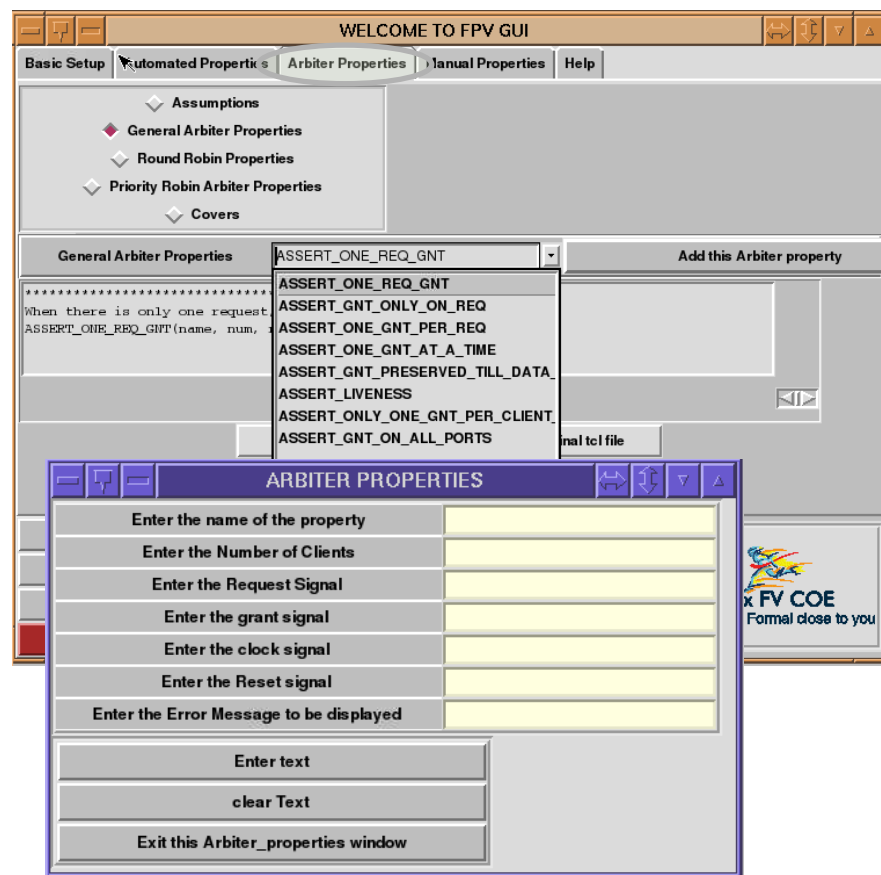
VPG FV Deployment Drive: Preparation Phase

- **Cookie-cutter Property Templates**
 - Automated properties
 - Standard Interfaces

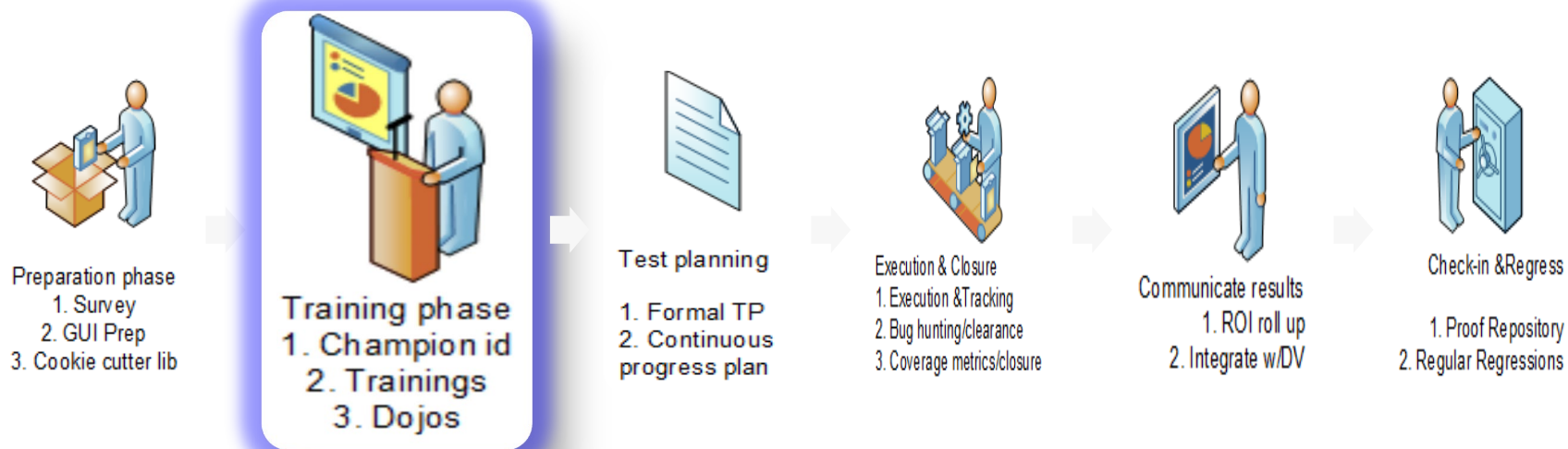


VPG FV Deployment Drive: Preparation Phase

- **Cookie-cutter Property Templates**
 - Automated properties
 - Standard Interfaces
 - Arbiter Properties



VPG FV Deployment Drive



-
- The diagram illustrates the formal verification process. It shows the flow from VCOEs (Verification Component Owners) and Formal Champions to the Central Formal Team. The Central Formal Team then interacts with the Setup, Constraints, Properties, Tools, GUIs, Methodologies, and Libraries. The process culminates in Formal Execution, which produces Reports to Management.
- ```

graph LR
 VCOEs1[VCOEs] --- FC1((Formal Champion))
 VCOEs2[VCOEs] --- FC2((Formal Champion))
 FC1 --- RTL[RTL]
 FC2 --- SCPS[Setup, Constraints, Properties]
 FC3((Formal Champion)) --- SCPS
 SCPS --- CFT[Central Formal Team]
 CFT --- TGLML[Tools, GUIs, Methodologies, Libraries]
 TGLML --- FE((Formal Execution))
 FE --- RMT[Reports to Management]

```

# VPG FV Deployment Drive



Preparation phase  
1. Survey  
2. GUI Prep  
3. Cookie cutter lib



Training phase  
1. Champion id  
2. Trainings  
3. Dojos



**Test planning**  
1. Formal TP  
2. Continuous progress plan



Execution & Closure  
1. Execution & Tracking  
2. Bug hunting/clearance  
3. Coverage metrics/closure



Communicate results  
1. ROI roll up  
2. Integrate w/DV

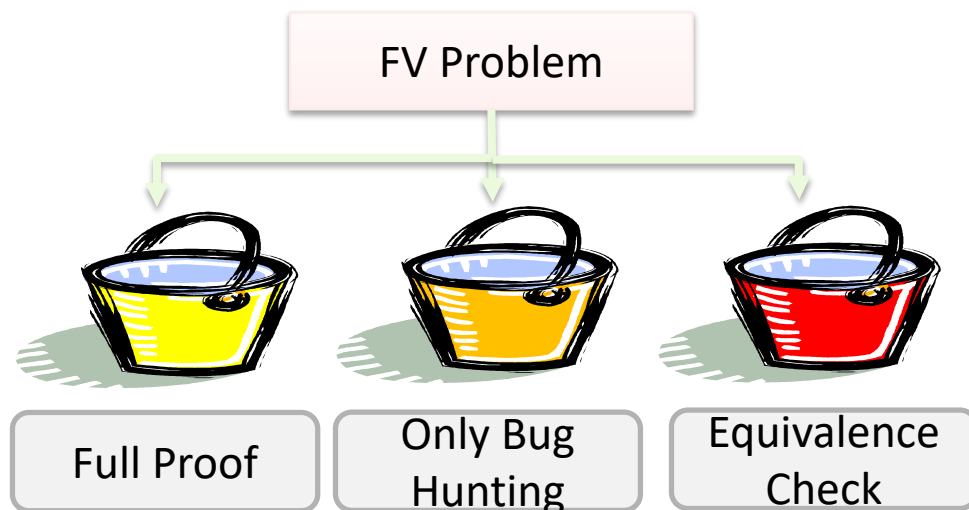


Check-in & Regress  
1. Proof Repository  
2. Regular Regressions

# VPG FV Deployment Drive:

## Test Planning Phase

- **Clear Goal Definitions**
  - Assigning FV problem to correct category





# VPF FV Deployment Drive:

## Test Planning Phase

- Progress Continuum**

- Defining measurable sub-goals
- Allows progress to be graded w.r.t. time
- Decided % completions to be achieved for each Val Milestone

| Over Constraint   | Checker set 1 | Checker set2 | Checker set3 | Checker set 4 | Checker set 5 |
|-------------------|---------------|--------------|--------------|---------------|---------------|
| Phase 1           | 1 week        |              |              |               |               |
| Phase 2           | 1.5 week      | 2 weeks      |              |               |               |
| Phase 3           | 2 weeks       | 2.5 weeks    | 3 weeks      |               |               |
| Valid Constraints | 3 weeks       | 3.5 weeks    | 4 weeks      | 5 weeks       | 6 weeks       |

# VPG FV Deployment Drive:

## Test Planning Phase

- **Activity Ownership**
  - Designer needs to monitor FV completion for his unit



# VPG FV Deployment Drive



Preparation phase  
1. Survey  
2. GUI Prep  
3. Cookie cutter lib



Training phase  
1. Champion id  
2. Trainings  
3. Dojos



Test planning  
1. Formal TP  
2. Continuous progress plan



Execution & Closure  
1. Execution & Tracking  
2. Bug hunting/clearance  
3. Coverage metrics/closure



Communicate results  
1. ROI roll up  
2. Integrate w/DV



Check-in & Regress  
1. Proof Repository  
2. Regular Regressions

# VPG FV Deployment Drive:

## Execution & Closure

- Execution and Tracking
  - Round-the-clock support extended to FV owners after dungeons
  - Weekly FV progress tracked and published
- Deep Bug Exploration
  - Abstractions, different engine settings
  - Lots of Covers added
- Coverage and Closure
  - Line and branch coverage obtained
  - Properties added for holes



# VPG FV Deployment Drive



# VPG FV Deployment Drive: Communicate Results

- ROI calculation standard was established
- ROI numbers calculated for activity and compared against DV

| ROI Parameter                   | Formal                     | Dynamic                    | Relative        |
|---------------------------------|----------------------------|----------------------------|-----------------|
| Bugs% found by FV               | $NB\_F/(NB\_F+NB\_D)$      | $NB\_D/(NB\_F+NB\_D)$      | $NB\_F/NB\_D$   |
| Total coverage achieved         | Covtot_F                   | Covtot_D                   |                 |
| Bug - engineering cost ROI      | $BEC\_F = NB\_F/CE\_F$     | $BEC\_D = NB\_D/CE\_D$     | $BEC\_F/BEC\_D$ |
| Bug - machine cost ROI          | $BMC\_F = NB\_F/Cm\_F$     | $BMC\_D = NB\_D/Cm\_D$     | $BMC\_F/BMC\_D$ |
| Coverage - engineering cost ROI | $CEC\_F = Covtot\_F/CE\_F$ | $CEC\_D = Covtot\_D/CE\_D$ | $CEC\_F/CEC\_D$ |
| Coverage - machine cost ROI     | $CmC\_F = Covtot\_F/Cm\_F$ | $CmC\_D = Covtot\_D/Cm\_D$ | $CmC\_F/CmC\_D$ |

| Parameter                                 | Depiction                       |
|-------------------------------------------|---------------------------------|
| Number of engineers worked on the problem | NE_F                            |
| Average bandwidth/engineer                | BW_F                            |
| Number of weeks spent on effort           | Ww_F                            |
| Total time spent in hours                 | $Ts\_F = BW\_F^a Ww\_Fa^a 40^b$ |
| Project schedule in weeks                 | Pw_F                            |
| Scaled engineering costs                  | $CE\_F = Ts\_F/(Pw\_F^a 40)$    |
| Machines used for computations            | Mc_F                            |
| Runs/week                                 | Rw_F                            |
| Number of days of run                     | Dw_F                            |
| Total machine-days/week                   | $Tm\_F = Mc\_F^a Rw\_F^a Dw\_F$ |
| Project machine-days/week                 | Pm_F                            |
| Scaled machine costs                      | $Cm\_F = Tm\_F/Pm\_F$           |
| Coverage (% cover points hit)             | Covtot_F                        |
| Bugs found by the method                  | NB_F                            |

<sup>a</sup>Assuming eight hours/day for a five-day work week = 40 hours/week.

<sup>b</sup>Machine runs are not limited to a working day but for the whole week

# VPG FV Deployment Drive



Preparation phase  
1. Survey  
2. GUI Prep  
3. Cookie cutter lib



Training phase  
1. Champion id  
2. Trainings  
3. Dojos



Test planning  
1. Formal TP  
2. Continuous progress plan



Execution & Closure  
1. Execution & Tracking  
2. Bug hunting/clearance  
3. Coverage metrics/closure



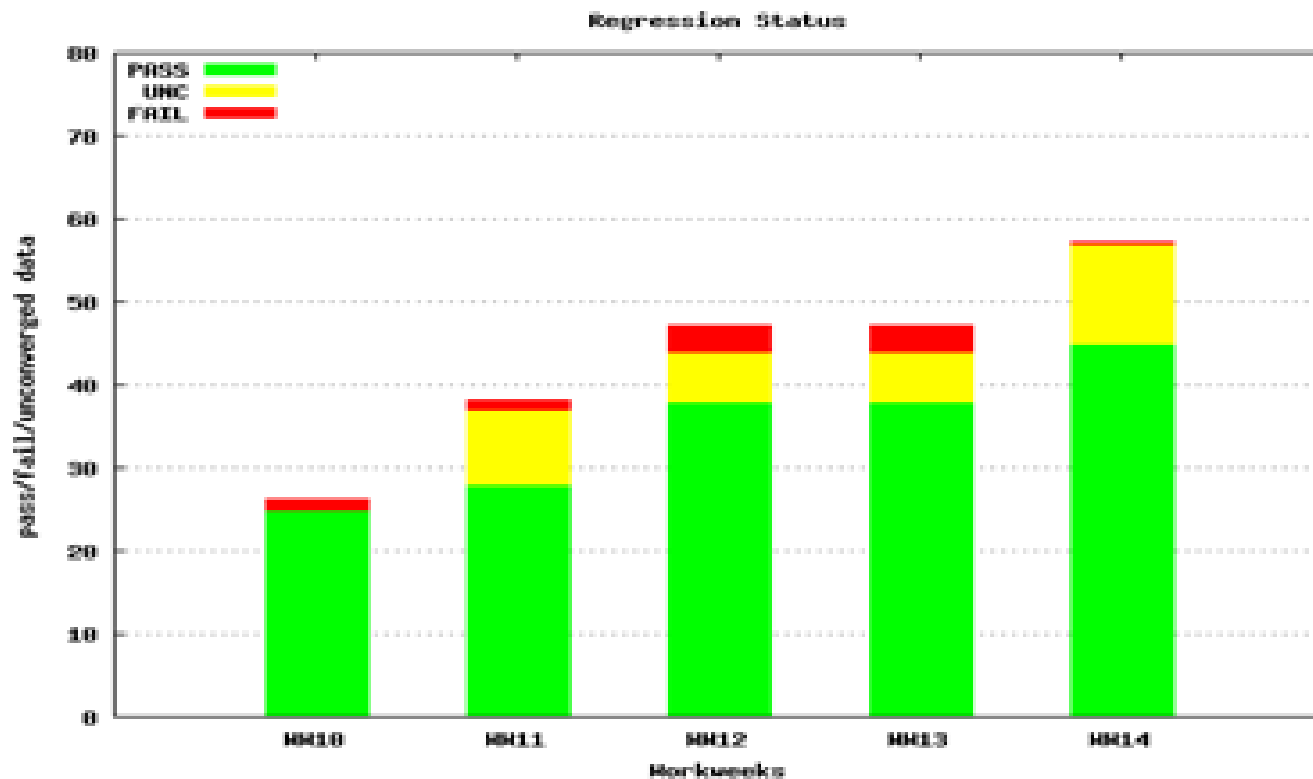
Communicate results  
1. ROI roll up  
2. Integrate w/DV



Check-in & Regress  
1. Proof Repository  
2. Regular Regressions

# VPG FV Deployment Drive: Check-In & Regress

- FV cannot be done once & forgotten
  - RTL in constant churn
  - Regressions to be enabled to find bugs due to changes





# AGENDA

Why?

- Need for FV Deployment Drive

How?

- VPG FV Deployment Planning and Execution

What was  
Achieved?

- Results of FV Deployment

# Results of VPG FV Deployment

- **Quality of reception:**

- 80+ engineers became FV experts
- 15 bugs caught within 2 weeks of activity

FV is awesome; I write code, run FV before even committing it to my local repo, and get it clean.

If it wasn't for this tool, I'd be building micro-testbenches before the ULT is ready.

It uncovered many scenarios in the beginning for which the Fulsim tests came very late. It has given very high confidence that new RTL has achieved the original health.

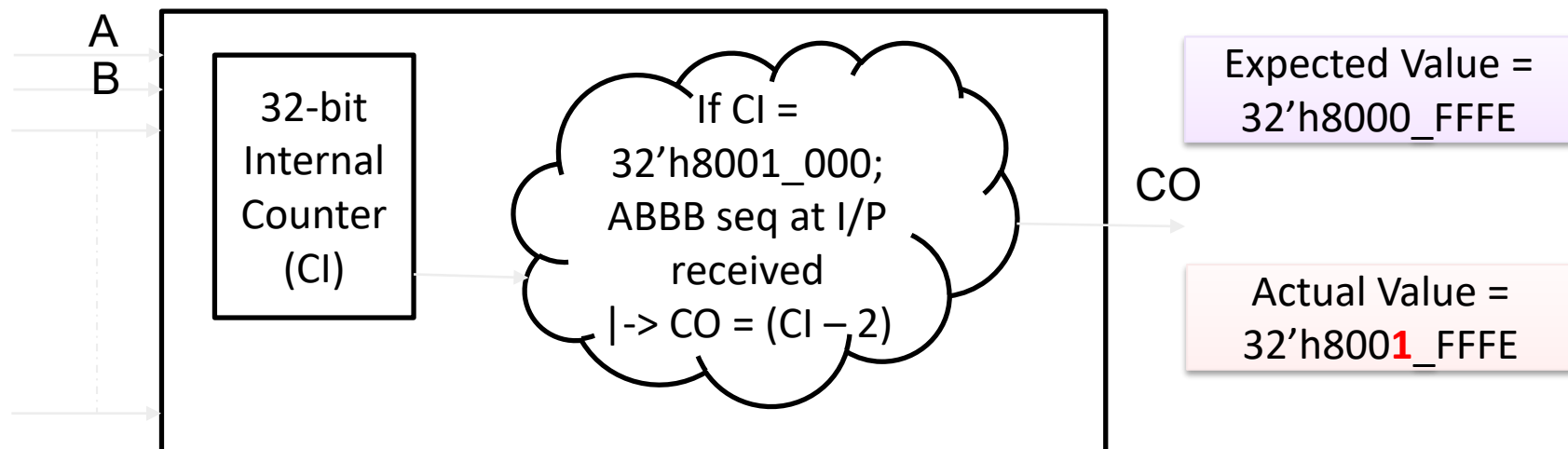
FV is the prime mode of validation for one of my units. While DV gave me a bill of 6-8 weeks, we verified all the DCN changes in less than 2 weeks using FV

Found couple of dozen bugs even before DV was up and helped clean the RTL

On a FV Quality bug hit:  
Hitting this bug in functional random env would have taken a long time and the bug is fatal enough to cripple the feature.

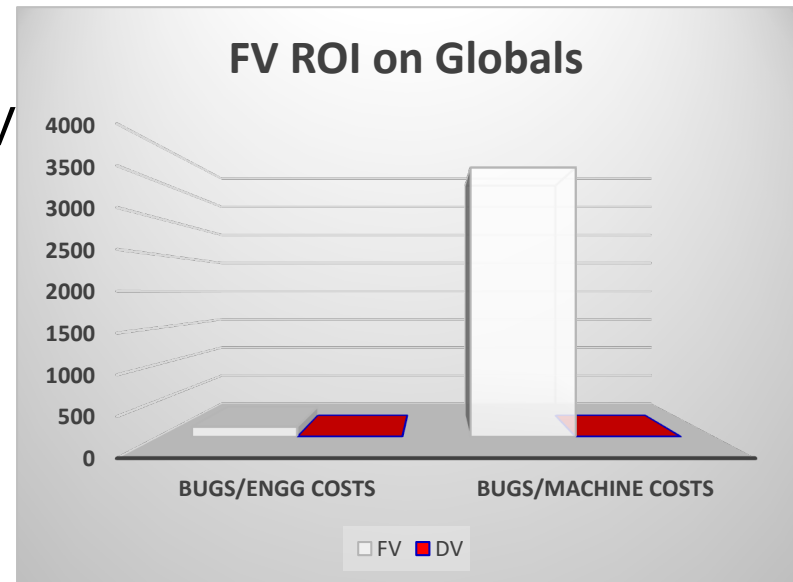
# Results of VPG FV Deployment

- **Quality of Bugs:**
  - 85+ bugs caught in 12 weeks (activity was started after half way through DV drive)
  - Most of the bugs caught fell in corner-case category



# Results of VPG FV Deployment

- **ROI:**
  - Weighted ROI of 220x compared to DV
  - Some activities gave as much as 138x w.r.t. engg costs; 3692x w.r.t. machine costs



# CONCLUSION

**FORMAL VERIFICATION IS NOW  
FORMALLY IN THE PROJECT!!**

# Q & A